Spring 5-25-2021

Classifying Illegal Advertisements on the Darknet Using NLP

Karan Shashin Shah

Classifying Illegal Advertisements on the Darknet Using NLP

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Karan Shashin Shah

May 2021

The Designated Project Committee Approves the Project Titled

Classifying Illegal Advertisements on the Darknet Using NLP

by

Karan Shashin Shah

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2021

Dr. Katerina Potika	Department of Computer Science
Dr. Robert Chun	Department of Computer Science
Dr. William Andreopoulos	Department of Computer Science

ABSTRACT

Classifying Illegal Advertisements on the Darknet Using NLP

by Karan Shashin Shah

The Darknet has become a place to conduct various illegal activities like child labor, contract murder, drug selling while staying anonymous. Traditionally, international and government agencies try to control these activities, but most of those actions are manual and time-consuming. Recently, various researchers developed Machine Learning (ML) approaches trying to aid in the process of detecting illegal activities. The above problem can benefit by using different Natural Language Processing (NLP) techniques. More specifically, researchers have used various classical topic modeling techniques like bag of words, N-grams, Term Frequency, Term Frequency Inverse Document Frequency (TF-IDF) to represent features and train machine learning models. Moreover, researchers have used an imbalanced dataset to perform those experiments.

In this work, we use some more modern techniques like Doc2Vec, Bidirectional Encoder Representation From Transformers (BERT) that have not been studied yet. The primary problem of this project is to classify illegal advertisements published on the Darknet by exploring the above-mentioned state of the art and comparing them against known approaches that use classical techniques, like TF-IDF. Also, we use various data balancing techniques and perform experiments using that data on classical techniques like TF-IDF.

Keywords - Darknet, Natural Language Processing (NLP), Term Frequency - Inverse Document Frequency (TF-IDF), Doc2Vec, Bidirectional Encoder Representation From Transformers (BERT)

ACKNOWLEDGMENTS

I like to thank my project advisor Dr. Katerina Potika for providing constant guidance, inspiration, and support. Throughout my research work, she constantly gave me directions to proceed further. I like to thank my committee members Dr. Robert Chun and Dr.William Andreopoulos, for their valuable feedback on the project. I like to thank G. Branwen for hosting the Darknet archives website and curating the darknet dataset prepared by various researchers through scrapping the dark market. Last but not least, I would like to thank the Almighty, my family, and my friends without whom none of this would have been possible.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

The Internet is a place where people enjoy the freedom of expressing their thoughts, sharing ideas, and speaking their minds. The only hindrance to this freedom is identity. A darknet is a place that removes this hindrance also. A darknet is a place known for its virtue of anonymity. In the darknet, users can enter the web world without getting tracked. Users can stay anonymous and perform all activities. In the darknet, all packets reach the destination after bouncing through various IP addresses. Hence, it gets very difficult to trace those packets and identify where they come from. Figure 1 shows the brief history of the darknet. Darknet was first formed in 1960 called ARPANET. It was only used by US defense. Darknet got open to the public in 2002

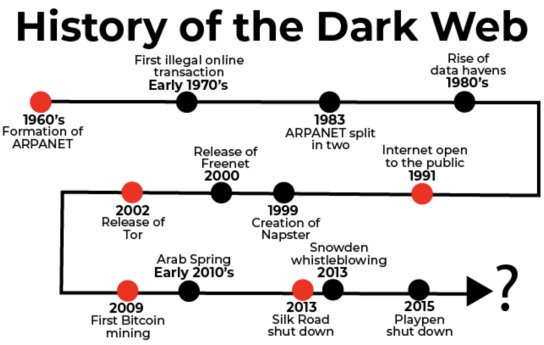


Figure 1: History of the dark web
[11]

With the darknet made public in 2002 and by an introduction to bitcoins in 2009, it has become a place to conduct various illegal activities. Content is primarily dominated by drugs-related materials, and other dubious objects and those things are advertised. Darknet has become a medium to conduct illicit activities like child labor, contract murder, etc. Traditional law enforcement agencies tried to control these activities by reconnaissance work or by following up on leads submitted by concerned

individuals. Both of the above-mentioned activities are not efficient and require lots of manual work [12]. Hence, classifying illegal activities has been a major issue in the darknet. With the increase in illegal activities, darknet called 'silk road' was shut down in 2009. But soon after, another dark marketplace was opened. Same users from 'silk road' started conducting the same illicit activities from the newly opened marketplace. Hence, user identification between different marketplaces is also another issue. To solve the issue, there was a need to implement machine learning algorithms. Many researchers have worked on solving these problems.

We use a dataset consisting of advertisements of products sold on two darknet marketplaces called Alphabay and Hansa, we are training machine learning models to classify the category of each product. Dataset is obtained from the darknet archives [13]. Researchers have done web scrapping of the various marketplaces and shared data on the website called darknet archives. Dataset consists of various features like item number, price, heading, product description, tags, feedback, product category, etc. Few advertisements of the dataset are not categorized. As we combine advertisements of two datasets, few advertisements of the same product are categorized with synonym names. We manually update the category of all such advertisements. In the end, all the advertisements are categorized into 13 categories which are Drugs and Chemicals, Carded Items, Jewels& Gold, Digital Products, Services, Fraud, Guides& Tutorials, Weapons, Software& Malware, Counterfeit items, Security & Hosting, Electronics, and Other Listings. Here Drugs and Chemicals, Carded Items, and Counterfeit items are illegal advertisements and the remaining are legal advertisements. Table 1 shows detail of the number of advertisements for each product

Using the above-processed dataset, we extract two features of the dataset, which are product description and product category, and try to solve the problem of classifying illegal advertisement by applying NLP techniques like Term Frequency Inverse

Table 1: Number of advertisements for each product

Classes	No of records
Drugs and Chemicals	83691
Fraud	13893
Digital Products	8914
Guides & Tutorials	7026
Counterfeit Items	3739
Services	3123
Other Listings	1407
Software & Malware	1089
Weapons	1083
Carded Items	796
Jewels & Gold	723
Security & Hosting	302
Electronics	31

Document Frequency (TF-IDF), Doc2Vec, and Bidirectional Encoder Representation from Transformers (BERT) to generate vectors and train machine learning model using that vectors.

CHAPTER 2

Definitions

Let us describe some basic definitions used in this project.

- Natural language processing (NLP) : It is branch of machine learning where we train models to understand human language.
- Darknet: It is network within internet which is accessed using specific software. People stay anonymous while exploring darknet.
- Doc2Vec : It is a NLP technique to represent each document as a vector which can be used to train machine learning models.
- Bidirectional Encoder Representations from Transformers (BERT) : It is machine learning technique for NLP based on transformers. It is pre trained by Google
- Corpus: It is collection of documents.
- Vocabulary: It is collection of words present in corpus.
- Bag of words : Text is considered as bag of words, without considering grammar and word order.
- Term Frequency(TF): It determines no of times a word appears in document.
- Inverse Document Frequency(IDF): It gives numerical value to a word which determines how important word is to a document or corpus.

2.1 Outcomes of any classification problem

There are possibly four outcomes of any classification problem.

- True Positive (TP): Model predicts particular class for an observation and it actually belongs to that class
- True Negative (TN): Model predicts an observation does not belong to particular class and it does not belong to that class.
- False Positive (FP): Model predicts particular class for an observation and it does not belong to that class

- False Negative (FN): Model predicts an observation does not belong to particular class and it does belong to that class.

2.2 Representation of outcomes of any classification problem

- ConfusionMetrics: Outcomes are plotted on the confusion metrics. Figure 2 is example of confusion metrics for binary classification

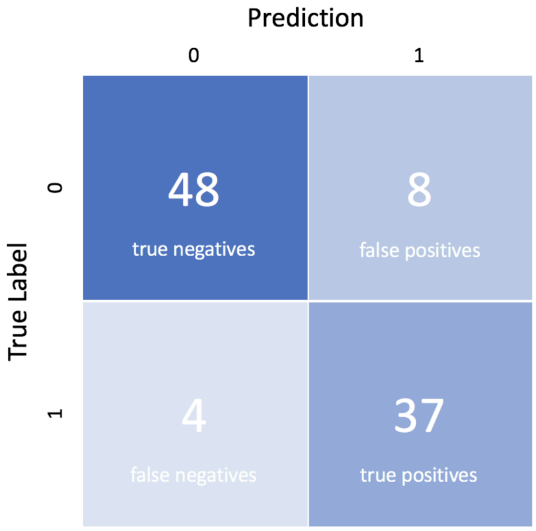


Figure 2: Example of ConfusionMetrics [1]

- Accuracy: Accuracy is total correct predictions divided by total predictions made by model on test data. Equation 1 shows how accuracy of model is calculated.

$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions} \tag{1}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

- Precision: Precision is defined as number of correct predictions made for particular class divided by total predictions made for particular class. Equation 3 shows how precision calculated for each class.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

- Recall: Recall is defined as number of correct predictions made for particular class divided by total number of observations that belong to that class. Equation 4 shows how recall is calculated for each class.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

- f1 score: F1 score is interpreted as average of precision and recall. Equation 5 shows how f1 score is calculated for each class.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

- Precision Recall curve (PR curve): Precision-Recall is considered useful metric to check success of model when dataset is highly imbalanced. Precision-recall curve is graph plotted where precision values are plotted on Y-axis and recall values are plotted on X-axis for every class in dataset. PR curve shows tradeoff between precision and recall.

  High precision and low recall signifies that predictions are made very few than actual for particular class but those predictions are correct. Low precision and high recall signifies that lots of observations are predicted for particular class but those predictions are not correct. High precision and high recall signify low false positve rate and low false negative rate. Figure 3 shows example for PR curve

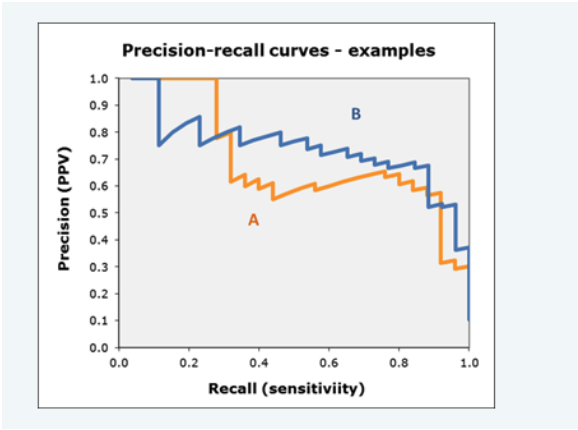


Figure 3: Example of PRCurve [2]

CHAPTER 3

Related Work

In 2011, G. Branwen [13] crawled silk road market place and published data on the darknet archives website. In 2011, Noor et al. [14] proposed a technique called "Query Probing" which is used to extract content from the dark web. Since then, researchers have worked extensively in classifying illegal advertisements on the darknet.

In 2012 Christin et al. [12] crawled silk road market place for 8 months and stated that large number of advertisements posted on the darknet are related to drugs. In 2014, Biryukov et. al. [15] performed classification of content on Tor hidden services and categorized them into 18 topics. Out of those topics, very few were illegal. Michael Graczyk et al. [16] used the dataset of darknet called Agora scrapped by G. Branwen [13] from June 6th, 2014 to July 7th, 2015. Michael Graczyk et al. used TF-IDF to form vectors from description features, performed Principal Component Analysis(PCA) to reduce feature dimensions. Finally trained Support Vector Machine(SVM) for classification of product into 12 classes and achieved accuracy of 79%. In 2016, Moore et. al. [17] developed Tor hidden services to extract data from the darknet. They collected 5k samples from tor onion pages and used SVM to classify products into 12 categories. In 2017, Ghosh et al. [18] prepared automated keyword extraction for product categories. They proposed a method of forming different term-frequency for each product category to form vectors. In 2017, Al Nabki et al. [19] created Darknet Usage Text Addresses(DUTA) dataset, manually classifying products into categories and subcategories. This dataset is considered very accurate and used by many researchers in future projects. In 2019, Al Nabki et al [20] used the same dataset which they prepared in 2017 and trained machine learning models to classify products into categories. They extended the DUTA dataset and

formed DUTA - 10K. They concluded that 20% of new data extracted was illegal. In 2019, Sagar Samtani et al. [21] used text mining to detect the high impact of opioid products. The latest research was around using Long Short Term Memory(LSTM) neural network [22] for classification of product advertisements but they have used dataset of product advertisements on Instagram.

Few researchers worked on identifying similar users in two or more darknets. S Shan et al. [23] made the hypothesis that accounts that are owned by the same individual across different marketplaces are likely to have the same usernames in all marketplaces. Susan et al. [24] also made a similar hypothesis and used profile images to identify similar users across different darknet after performing image analysis.

Most of the researchers used TF-IDF [19] or Bag Of Words (BOW) [25] to form vectors of product description and trained SVM, K Nearest Neighbour(KNN), Logistic Regression and Naïve Bayes to classify advertisement to product category. Feature reduction is made by most of the researchers through PCA [16] and Latent Dirichlet Allocation (LDA).

Our work is an attempt to use the state of arts like Doc2Vec and BERT to classify illegal advertisements on the darknet. Apart from that, we tried various data balancing techniques on the dataset and formed vectors from the dataset using the already explored TF-IDF technique to train machine learning models like SVM, KNN, and Naïve Bayes.

CHAPTER 4

Problem Definition

As mentioned in the previous section, we are converting problem of identifying illegal advertisements on darknet to NLP problem. The most important step of any NLP problem is vectorization. We form vectors of product description feature using various vectorization techniques like TF-IDF, Doc2Vec and BERT. In this section, we will explain each of these vectorization techniques in detail.

4.1 TF-IDF

Term Frequency (TF) is count of each word appearing in document. Algorithm keeps track of each word that appears in document and generate vector for each document. Table 2 shows how TF is calculated

$$tf(t,d) = \frac{f_{t,d}}{\sum f_{t',d}} \tag{6}$$

t = tokens d = document $f_{t,d}$ = no of tokens t in document d

Inverse document frequency (idf) checks word across set of documents. It checks whether given word is common across all the documents. Idf value closer to 0 for a particular word signifies that word is common across all the documents. Below is the formula to calculate idf

$$idf(t,D) = \log \frac{N}{d \epsilon D : t \epsilon d} \tag{7}$$

N = total number of documents t = token d = document D = all the documents

Table 2: Term Frequency (TF)

	You	Are	Who
You Are	1	1	0
Who Are You	1	1	1

Tf-Idf is the product of term frequency and inverse document frequency. Below is the equation to find Tf-Idf

$$tf - idf(t, d) = tf(t, d) \times idf(t, D) \tag{8}$$

Here, we take product description of each advertisement as document and computed tf-idf. Vectors are generated for product description of each advertisement using tf-idf.

4.2 Doc2Vec

The biggest issue of TF-IDF technique is requirement of large amount of memory. In TF-IDF technique, dimensions of vector is equal to number of unique words in corpus which is very high and requires large amount of memory. To solve this problem, Mikolov et al. [26] came up with idea of network-based word representation called Word2Vec.

Suppose, we are given words $w_1, w_2, ..., w_n$, Word2Vec maximizes predicted log probability. Equation 9 determines predicted log probability.

$$\frac{1}{T} \sum_{T-k}^{t-k} \log p(w_t | w_{t-k}, ..., w_{t+k}) \tag{9}$$

where k = window size for preserving the contextual information

Softmax function does prediction as mentioned in Equation 10

$$p(w_t | w_{t-k}, ..., w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{10}$$

where y_i is the ith output value of feed forward neural network. Feed forward neural network is computed using Equation 11

$$y = b + Uh(w_{t-k}, ..., w_{t+k}; W) \tag{11}$$

where b is bias between the hidden and output layer, U is weight matrix between the hidden and output layer, h is average for context words, W is word embedding

matrix

Doc2Vec is extension of Word2Vec. The only addition in Doc2Vec is each document is mapped to document vector which is at same space of word vectors. Hence, for Doc2Vec, there will be addition of D in the Equation 11 as follows:

$$y = b + Uh(w_{t-k}, ..., w_{t+k}; W) \tag{12}$$

Figure 4 shows difference between Word2Vec and Doc2Vec embeddings.

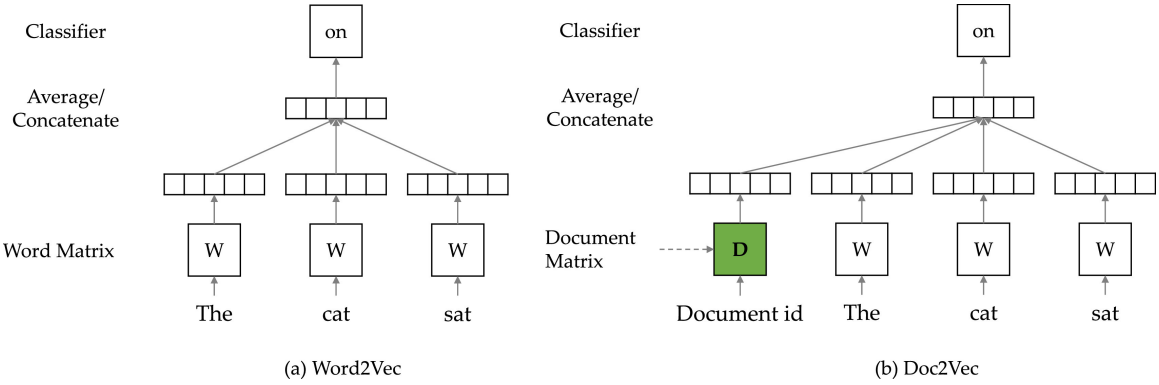


Figure 4: word2VecvsDoc2Vec Architecture [3]

Two primary structures of Doc2Vec are PV-DM and PV-DBOW.

4.3 BERT

BERT stands for Bidirectional Encoder Representation Transformer. Google AI researchers pre -train deep bidirectional representation of words using unlabeled text. By adding one output layer to that pre trained model, various NLP problems like text classification can be solved.

BERT consists of L identical transformer encoder layers. Each of these layers contain two types of sublayer. First layer is multi-head self-attention mechanism. This layer encodes specific word and also look at other words in sequence to derive contextual meaning. Second layer is fully connected feed-forward network (FFN). This consists of two linear transformations. They are

$(W_1 \varepsilon R^{d_{model} \times d_{ff}}, b_1 \varepsilon R^{d_{ff}}), (W_2 \varepsilon R^{d_{model} \times d_{ff}}, b_2 \varepsilon R^{d_{ff}})$ such that

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{13}$$

FFN uses GELU activation which is defined as

$$GELU(x) = 0.5x(1 + tanh(\sqrt{2/\prod}(x + 0.044715x^3))) \tag{14}$$

Each encoder layer has residual connection and layer normalization such that output of each sublayer is

$$LayerNorm(x + Sublayer(x)) \tag{15}$$

Figure 5 shows architecture of BERT.

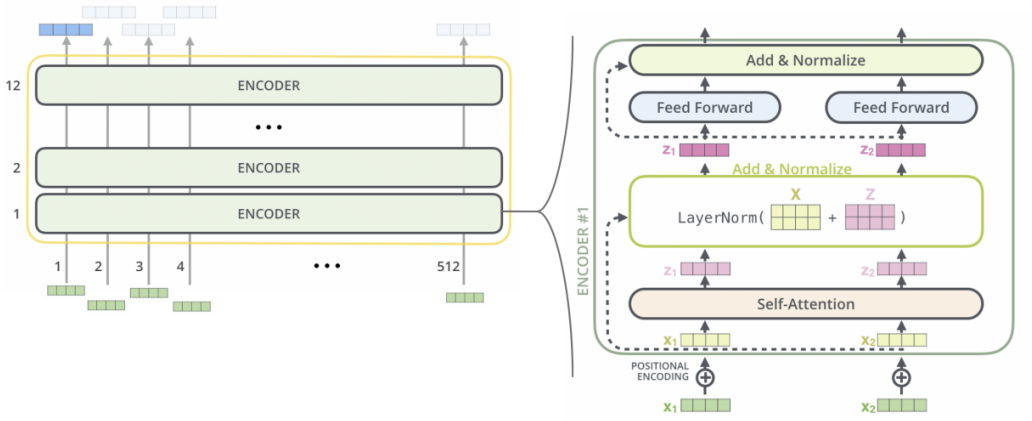


Figure 5: BERT architecture [4]

Figure 6 shows versions of BERT

First version is BERT-base: L=12, $d_{model} = 768$, h=12, $d_{ff} = 3072$ (110M total parameters). Second version is BERT-base: L=24, $d_{model} = 1024$, h=16, $d_{ff} = 4096$ (340M total parameters).

Where L is number of layers, d_{model} is the dimensionality of input and output of each layer, h is the number of attentions heads in a self-attention sublayer and d_{ff} is the number of hidden units in feed-forward sublayer.

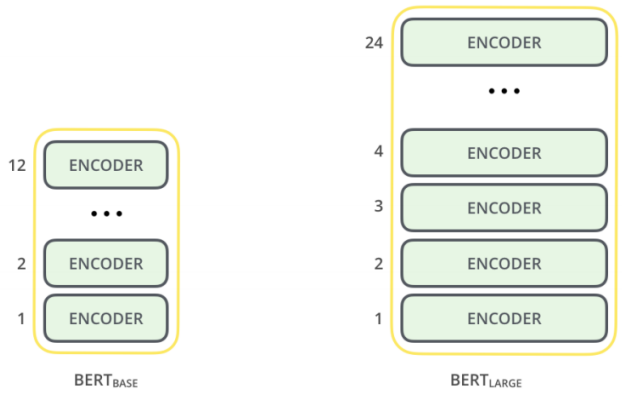


Figure 6: BERT Version [4]

CHAPTER 5

Datasets and Methods

5.1 Dataset

For our experiment, we use Alphabay dataset consisting of $114,231$ records and 13 columns. Table 3 shows features of Alphabay dataset. Another dataset which we use is Hansa marketplace dataset consisting of $12,173$ records and 7 columns. Table 4 shows features of Hansa dataset. Both the datasets are available on darknet archives.

Table 3: Alphbay dataset's features

itemnumber	sold	product	listed	origin	shipsto	quantity
productdescription	class	Heading	vendor level	trustlevel	tags	category

Table 4: hansa marketplace dataset's features

Date	Number	Product Description	Handle	Currency	Category	Country

We extract two features called product description and category from each dataset and form final dataset consisting of 2 columns and 126404 records. Few records are not categorized in Hansa marketplace dataset and similar category records are named with synonymous name. We correct all those manually to form the final dataset. Table 4 shows features of the final dataset.

Table 5: final dataset's features

Product Description	Category

5.2 Data pre-processing

In NLP problems, data pre-processing is extremely important to get higher accuracy from trained machine learning model. Figure 7 shows the complete data pre-processing pipeline.

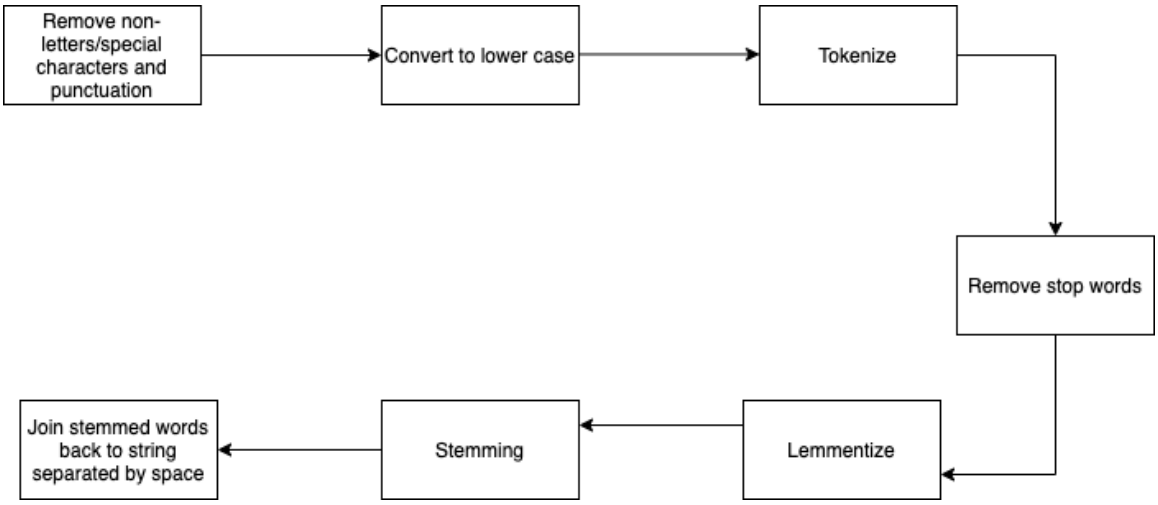


Figure 7: Data pre-processing pipeline

5.2.1 Remove non-letters/special characters and punctuation

We remove all non-letters, special characters and punctuation from product description feature.

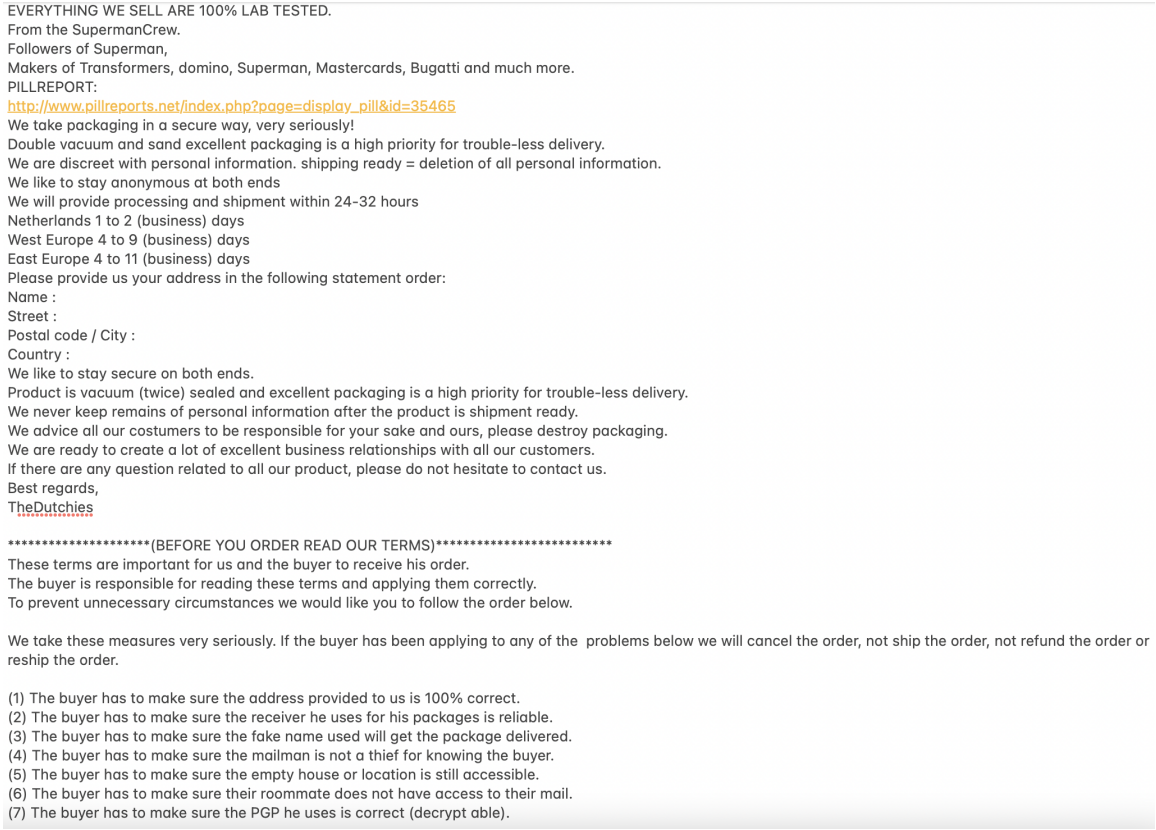
5.2.2 Convert to lower case

Since our dataset is in english language, it has two cases. Upper case and lower case. In this step, we convert every letter to lower case since we are interested in capturing only semantic meaning of word.

5.2.3 Tokenization

Tokenization is the fundamental data pre-processing step for any NLP problem. There are various tokenization techniques. We implement the basic tokenization technique which is called word tokenization. Word tokenizer splits text of the entire document into words separated by certain delimiters. We apply word tokenization on

each product description record separated by space delimeter and store them in new column called tokens in dataset. Figure 8 and Figure 9 shows one product description record and its corresponding tokens respectively.



EVERYTHING WE SELL ARE 100% LAB TESTED.
From the SupermanCrew.
Followers of Superman,
Makers of Transformers, domino, Superman, Mastercards, Bugatti and much more.
PILLREPORT:
http://www.pillreports.net/index.php?page=display_pill&id=35465
We take packaging in a secure way, very seriously!
Double vacuum and sand excellent packaging is a high priority for trouble-less delivery.
We are discreet with personal information. shipping ready = deletion of all personal information.
We like to stay anonymous at both ends
We will provide processing and shipment within 24-32 hours
Netherlands 1 to 2 (business) days
West Europe 4 to 9 (business) days
East Europe 4 to 11 (business) days
Please provide us your address in the following statement order:
Name :
Street :
Postal code / City :
Country :
We like to stay secure on both ends.
Product is vacuum (twice) sealed and excellent packaging is a high priority for trouble-less delivery.
We never keep remains of personal information after the product is shipment ready.
We advice all our costumers to be responsible for your sake and ours, please destroy packaging.
We are ready to create a lot of excellent business relationships with all our customers.
If there are any question related to all our product, please do not hesitate to contact us.
Best regards,
TheDutchies

**********(BEFORE YOU ORDER READ OUR TERMS)**********
These terms are important for us and the buyer to receive his order.
The buyer is responsible for reading these terms and applying them correctly.
To prevent unnecessary circumstances we would like you to follow the order below.

We take these measures very seriously. If the buyer has been applying to any of the problems below we will cancel the order, not ship the order, not refund the order or reship the order.

(1) The buyer has to make sure the address provided to us is 100% correct.
(2) The buyer has to make sure the receiver he uses for his packages is reliable.
(3) The buyer has to make sure the fake name used will get the package delivered.
(4) The buyer has to make sure the mailman is not a thief for knowing the buyer.
(5) The buyer has to make sure the empty house or location is still accessible.
(6) The buyer has to make sure their roommate does not have access to their mail.
(7) The buyer has to make sure the PGP he uses is correct (decrypt able).

Figure 8: Product description of one advertised product.

['EVERYTHING', 'WE', 'SELL', 'ARE', '100', '%', 'LAB', 'TESTED', '.', 'From', 'the', 'SupermanCrew', '.', 'Followers', 'of', 'Superman', ',', 'Makers', 'of', 'Transformers', ',', 'domino', ',', 'Superman', ',', 'Mastercards', ',', 'Bugatti', 'and', 'much', 'more', '.', 'PILLREPORT', ':', 'http', ':', '//www.pillreports.net/index.php', '?', 'page=display_pill', '&', 'id=35465', 'We', 'take', 'packaging', 'in', 'a', 'secure', 'way', ',', 'very', 'seriously', '!', 'Double', 'vacuum', 'and', 'sand', 'excellent', 'packaging', 'is', 'a', 'high', 'priority', 'for', 'trouble-less', 'delivery', '.', 'We', 'are', 'discreet', 'with', 'personal', 'information', '.', 'shipping', 'ready', '=', 'deletion', 'of', 'all', 'personal', 'information', '.', 'We', 'like', 'to', 'stay', 'anonymous', 'at', 'both', 'ends', 'We', 'will', 'provide', 'processing', 'and', 'shipment', 'within', '24-32', 'hours', 'Netherlands', '1', 'to', '2', '(', 'business', ')', 'days', 'West', 'Europe', '4', 'to', '9', '(', 'business', ')', 'days', 'East', 'Europe', '4', 'to', '11', '(', 'business', ')', 'days', 'Please', 'provide', 'us', 'your', 'address', 'in', 'the', 'following', 'statement', 'order', ':', 'Name', ':', 'Street', ':', 'Postal', 'code', '/', 'City', ':', 'Country', ':', 'We', 'like', 'to', 'stay', 'secure', 'on', 'both', 'ends', '.', 'Product', 'is', 'vacuum', '(', 'twice', ')', 'sealed', 'and', 'excellent', 'packaging', 'is', 'a', 'high', 'priority', 'for', 'trouble-less', 'delivery', '.', 'We', 'never', 'keep', 'remains', 'of', 'personal', 'information', 'after', 'the', 'product', 'is', 'shipment', 'ready', '.', 'We', 'advice', 'all', 'our', 'costumers', 'to', 'be', 'responsible', 'for', 'your', 'sake', 'and', 'ours', ',', 'please', 'destroy', 'packaging', '.', 'We', 'are', 'ready', 'to', 'create', 'a', 'lot', 'of', 'excellent', 'business', 'relationships', 'with', 'all', 'our', 'customers', '.', 'If', 'there', 'are', 'any', 'question', 'related', 'to', 'all', 'our', 'product', ',', 'please', 'do', 'not', 'hesitate', 'to', 'contact', 'us', '.', 'Best', 'regards', ',', 'TheDutchies', '**********', '(', 'BEFORE', 'YOU', 'ORDER', 'READ', 'OUR', 'TERMS', ')', '**********', 'These', 'terms', 'are', 'important', 'for', 'us', 'and', 'the', 'buyer', 'to', 'receive', 'his', 'order', '.', 'The', 'buyer', 'is', 'responsible', 'for', 'reading', 'these', 'terms', 'and', 'applying', 'them', 'correctly', '.', 'To', 'prevent', 'unnecessary', 'circumstances', 'we', 'would', 'like', 'you', 'to', 'follow', 'the', 'order', 'below', '.', 'We', 'take', 'these', 'measures', 'very', 'seriously', '.', 'If', 'the', 'buyer', 'has', 'been', 'applying', 'to', 'any', 'of', 'the', 'problems', 'below', 'we', 'will', 'cancel', 'the', 'order', ',', 'not', 'ship', 'the', 'order', ',', 'not', 'refund', 'the', 'order', 'or', 'reship', 'the', 'order', '.', '(', '1', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'address', 'provided', 'to', 'us', 'is', '100', '%', 'correct', '.', '(', '2', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'receiver', 'he', 'uses', 'for', 'his', 'packages', 'is', 'reliable', '.', '(', '3', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'fake', 'name', 'used', 'will', 'get', 'the', 'package', 'delivered', '.', '(', '4', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'mailman', 'is', 'not', 'a', 'thief', 'for', 'knowing', 'the', 'buyer', '.', '(', '5', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'empty', 'house', 'or', 'location', 'is', 'still', 'accessible', '.', '(', '6', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'their', 'roommate', 'does', 'not', 'have', 'access', 'to', 'their', 'mail', '.', '(', '7', ')', 'The', 'buyer', 'has', 'to', 'make', 'sure', 'the', 'PGP', 'he', 'uses', 'is', 'correct', '(', 'decrypt', 'able', ')', '.']

Figure 9: tokens formed of product description

5.2.4 Removing stop words

There are few words which are common across all the documents and add very little to no value in classifying the given document. Those words should be removed so that important words are used to form vector that represents document. Vector space model formed after removing stop words help train machine learning model in better way. Figure 10 shows list of 25 stop words of English language

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

Figure 10: 25 stop words list of English language [5]

5.2.5 Lemmatization

There are various word normalization techniques in NLP. In any language, there are lots of words which are derived from one another. These words are called inflected words. Figure 11 shows example of word normalization.

Word normalization is converting these derived words to corresponding root word. Lemmatization is one such word normalization technique. It converts all derived words to single root word in document.

We apply lemmatization on tokens generated in previous step.

5.2.6 Stemming

Stemming is another word normalization technique. In this method, common prefix and suffix are removed from the word to convert them into their corresponding root word.Table 6 shows example of stemming.

5.2.7 Join words back into one string separted by space

In this step, we join all tokens into one string for each product description record.

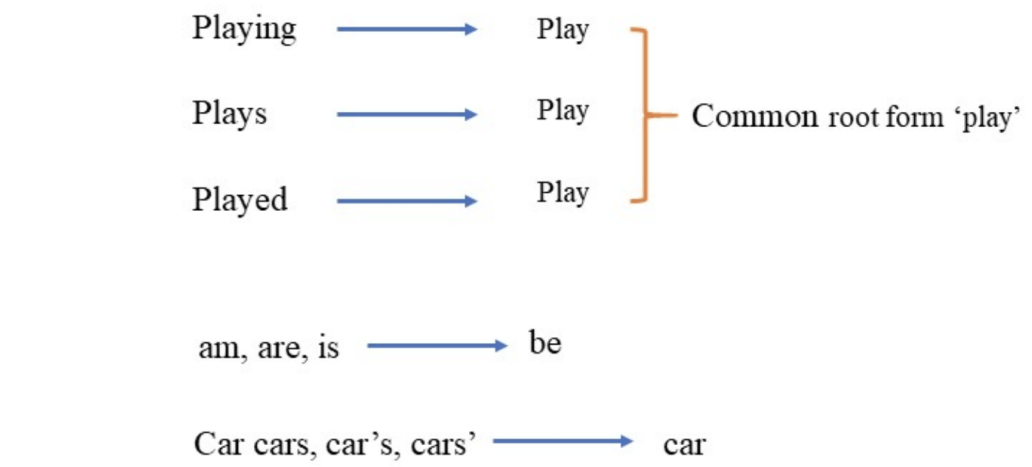


Figure 11: word normalization example [6]

Table 6: Stemming example

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study

5.2.8 Removing null valued records

After performing all the above mentioned steps of data pre-processing, few tokens value become null. We are removing all those records whose token value gets null after data pre-processing.

After performing all the above mentioned steps of data pre-processing, size of the document is reduced. Figure 12 and Figure 13 are code snippets describing size of document before and after data pre processing steps respectively.

```
[11] print(len(finalDataset['Product Description'][0]))

     2455
```

Figure 12: Document length before data pre-processing

```
print (len(token[0]))

2198
```

Figure 13: Document length after data pre-processing

5.3 Balancing dataset

As evident in Table 1, dataset is skewed towards the advertisements of class Drugs and Chemicals. This makes dataset imbalanced and biased which in turn affects the training of model.

We apply Synthetic minority oversampling technique (SMOTE) and under sampling technique to balance the dataset.

5.3.1 SMOTE analysis

In this approach, minority class data is over sampled. One way is to generate new data points by duplicating minority class data. But this way does not add any information to the data. This increases chances of overfitting while training our model. Another way is SMOTE analysis. In this method, examples which are close in feature space are selected and a line is drawn between these examples in feature space and new samples are created along that line.

5.3.2 Under Sampling

This is another approach of balancing the dataset. In this approach, majority class data is removed to balance the dataset. Although, dataset is getting balanced using this approach, valuable information is getting lost.

Figure 14 shows my overall implementation approach.

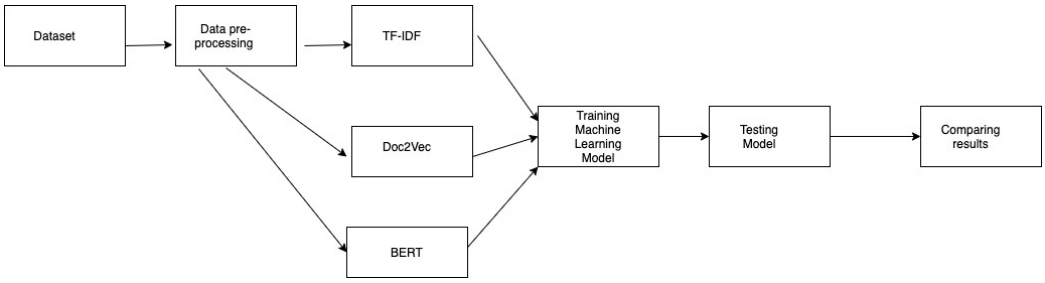


Figure 14: Overall approach for our classification problem

5.4 Machine Learning Algorithm

We solved this classification problem using supervised machine learning approach. Supervised learning approach is where class labels are given in dataset and we have to train machine learning model to classify input in one of those labels.

5.4.1 Support Vector Machine (SVM)

SVM is supervised machine learning approach where algorithm is trained to form hyperplanes for each class. While training model, vectors formed from the dataset are taken to higher dimensions and get plotted in any one of the hyperplane which determines its class.

5.4.2 K Nearest Neighbors (KNN)

KNN is simple supervised machine learning approach. K nearest data points determine the class of data point. For example, K = 3 and two nearest data points are labelled class A and one data point is labelled class B then given data point is classified as class A.

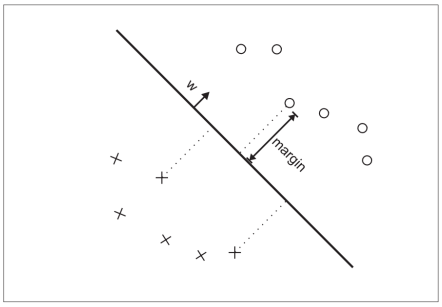


Figure 15: Support Vector Machine Example [7]

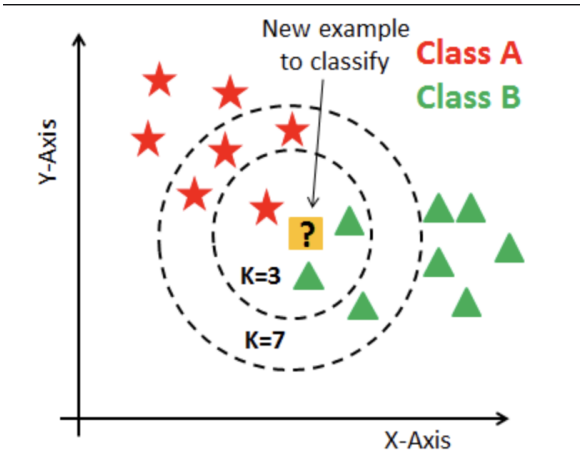


Figure 16: K Nearest Neighbour Example [8]

5.4.3 Naïve Bayes

Naïve Bayes is a machine learning approach based on Bayes Theorem. It assumes that all variables are independent of each other which is not true in real world scenario yet, this algorithm gives great result in problems like text classification. Baye's theorem states following relationship given class variable y and dependent feature vector x_1 to x_n.

$$P(y|x_1.....x_n) = \frac{P(y)P(x_1.....x_n|y)}{P(x_1.....x_n)} \tag{16}$$

Using Naïve Bayes assumption

$$P(y|x_1.....x_n) = P(x_i|y, x_1, x_2, ..., x_i, x_{i+1}, ...x_n) = P(x_i|y) \tag{17}$$

for all i,

$$P(y|x_1....x_n) = \frac{p(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1.....x_n)} \tag{18}$$

Since P$(x_1, ..., x_n)$ is constant given the input, we can use following classification rule

for all i,

$$P(y|x_1....x_n) = P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{19}$$

$$P(y|x_1....x_n) = \hat{y} = argmaxP(y)\prod_{i=1}^{n} P(x_i|y), \tag{20}$$

and we use Maximum A Posteriori (MAP) estimation to estimate P(y) and P$(x_i|y)$

There are various variants of Naïve Bayes algorithms. Multinomial Naive Bayes is classic variant primarily used for text classification. We traine Multinomial Naive Bayes for our problem.

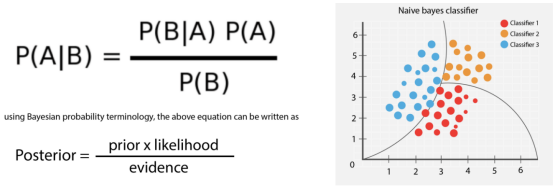


Figure 17: NaïveByes Example [9]

5.4.4 Logistic Regression

Logistic regression is simple machine learning model. It is a regression analysis. It predicts class for each data point.

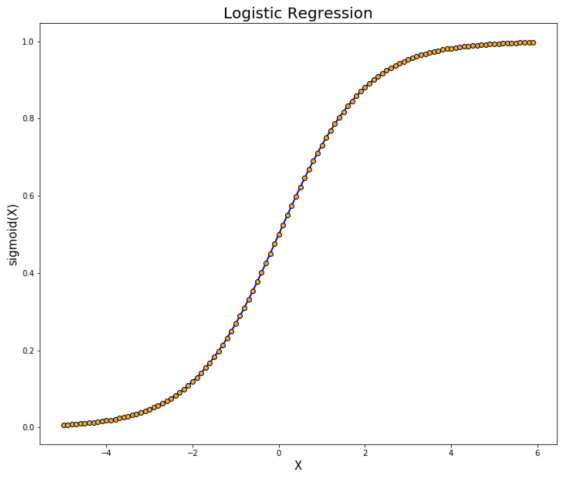


Figure 18: LogisticRegression Example [10]

5.5 Evaluation Metrics

We evaluate our experiments through various measures like accuracy, precision, recall, PR curve, Average Precision(AP) score.

CHAPTER 6

Experimental Results

6.1 Experiment Setup

Python is used as implementation language. Python libraries like pandas, sklearn, seaborn, pickle, nltk, huggingface are used for data pre-processing , training and testing model. Microsoft excel is used to prepare dataset. Cloud service - google colab with GPU is used for compiling and running code. All experiments are conducted on a MacBook with macOS Big Sur version 11.2.3.

6.2 Results

This section contains results of various experiments performed to solve problem.

6.2.1 TF-IDF

After generating vectors using TF-IDF technique, we train SVM, KNN and Naïve Bayes machine learning models.

6.2.1.1 Support Vector Machine(SVM) Results

We experiment to train SVM using various kernel functions, regularization and degree. Table 7 shows accuracy of SVM for different hyper parameters.

As evident from Table 7, SVM with linear kernel function, regularization parameter 1 and degree 3 give the highest accuracy. Further analysis is done with SVM model trained using linear function, regularization parameter 1 and degree 3.

Table 7: SVM results

Kernel	Regularization (C)	Degree	Accuracy(%)
Linear	1.0	3	86.50
Linear	1.0	1	85.69
rbf	1.0	2	66.54
rbf	1.0	3	66.70
poly	1.0	2	66.70
Linear	3.0	3	86.11

Figure 19 shows confusion matrix, Figure 20 shows micro averaged precision score of all classes, Figure 21 shows PR value for each class from results obtained by testing SVM model trained on imbalanced dataset.

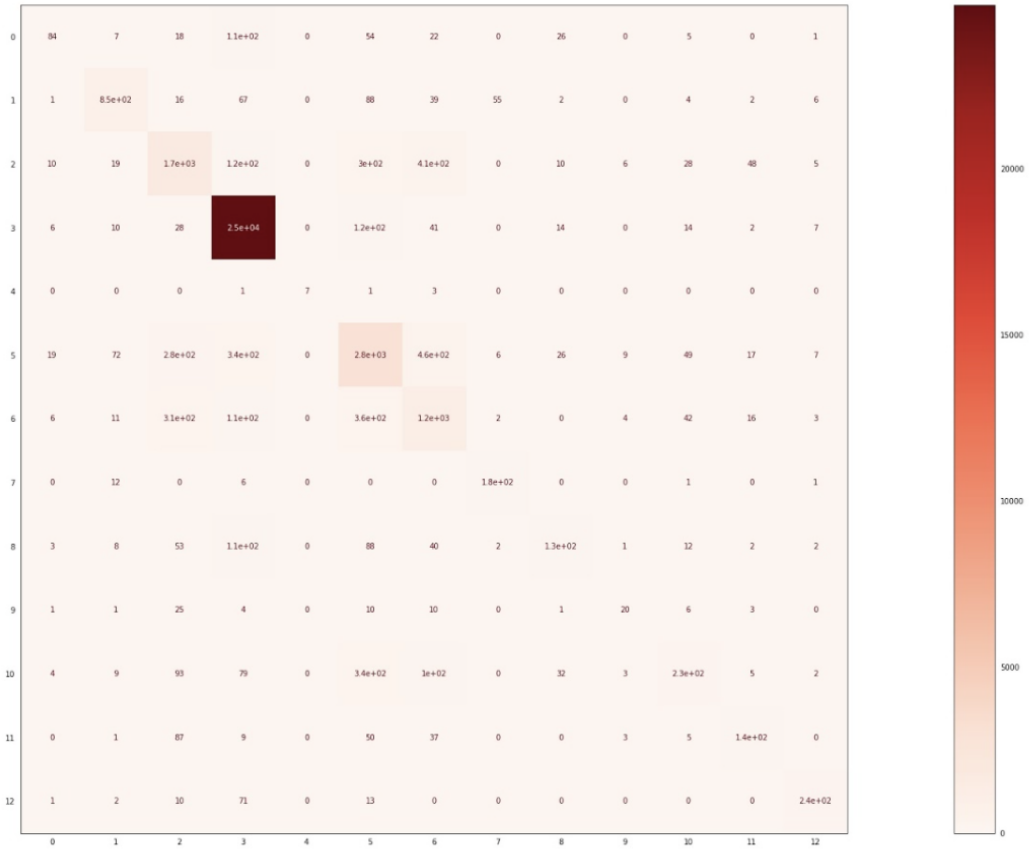


Figure 19: Confusion Matrix obtained from result of SVM trained on imbalanced dataset

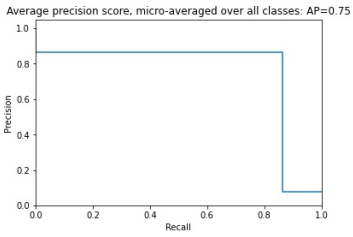


Figure 20: Micro averaged precision score obtained from the results of SVM trained on imbalanced dataset

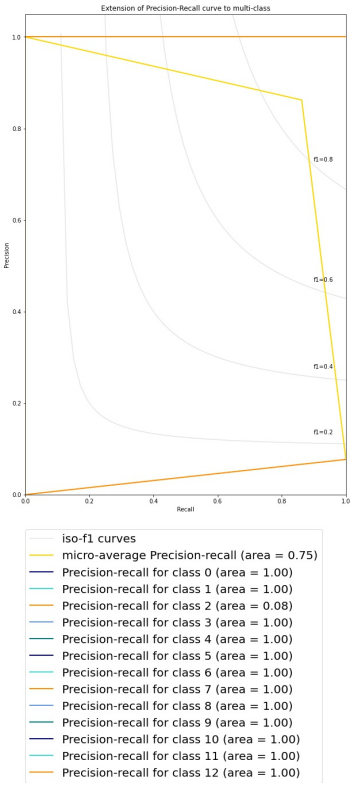


Figure 21: PR value obtained for each class from result of SVM trained on imbalanced dataset

Table 8: KNN results

K	Accuracy (%)
3	74.54
4	78.49
5	78.27
6	77.86

6.2.1.2 K Nearest Neighbour(KNN) Results

Although learning is not involved in KNN machine learning model, KNN gives high result in text classification and semantic analysis problem. We experiment using different values of K to train KNN and Table 8 shows accuracy for the same.

As evident from Table 8, model with K value 4 gives the highest accuracy. Further

analysis is done for the results obtained from trained KNN model with K value 4.

Figure 22 shows confusion matrix, Figure 23 shows micro averaged precision score of all classes, Figure 24 shows PR value for each class from results obtained by testing KNN model trained on imbalanced dataset.

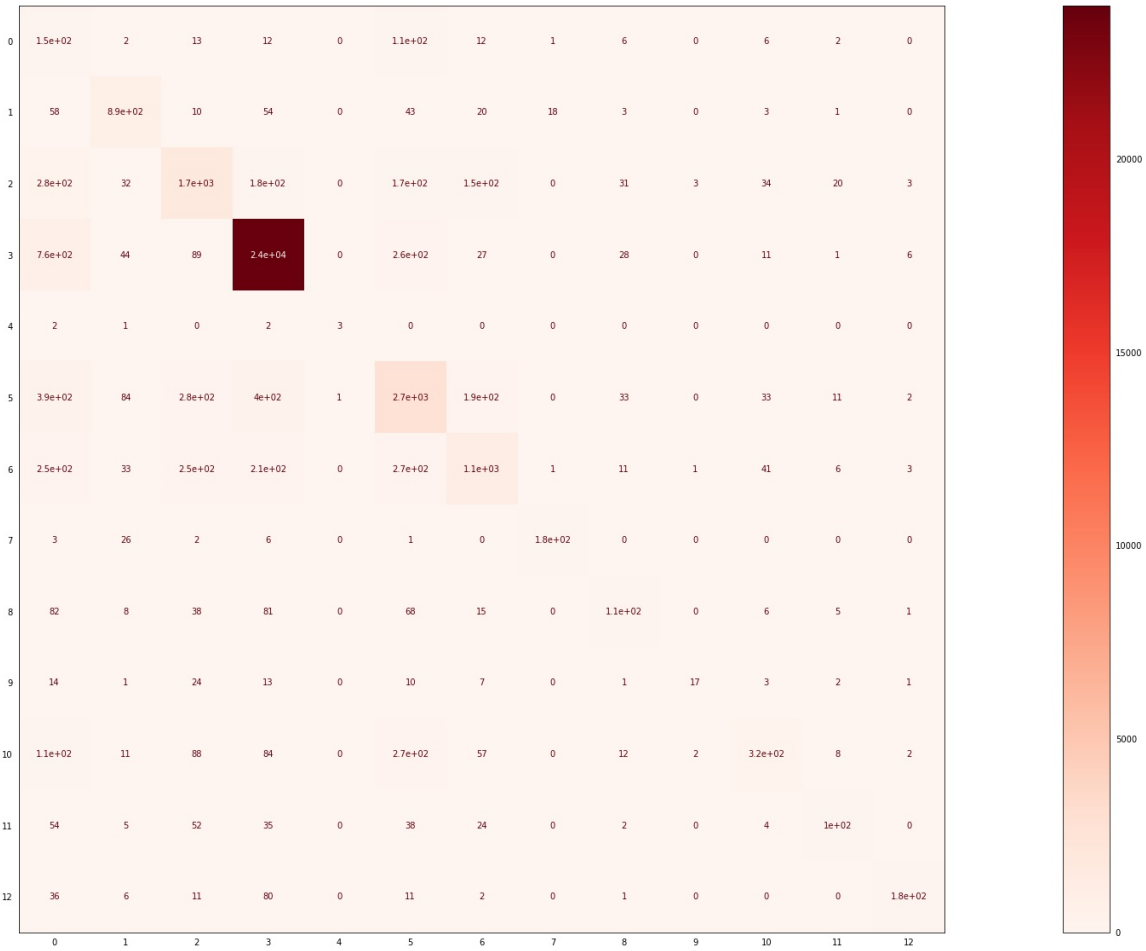


Figure 22: Confusion matrix obtained from result of KNN trained on imbalanced dataset

As class is highly imbalanced, we balance the dataset using various data balancing technique.

We perform SMOTE analysis on dataset for upsampling and achieve accuracy of 73.01%. Figure 25 shows confusion matrix, Figure 26 shows micro averaged precision

Figure 23: Micro averaged precision score obtained from results of KNN trained on imbalanced dataset

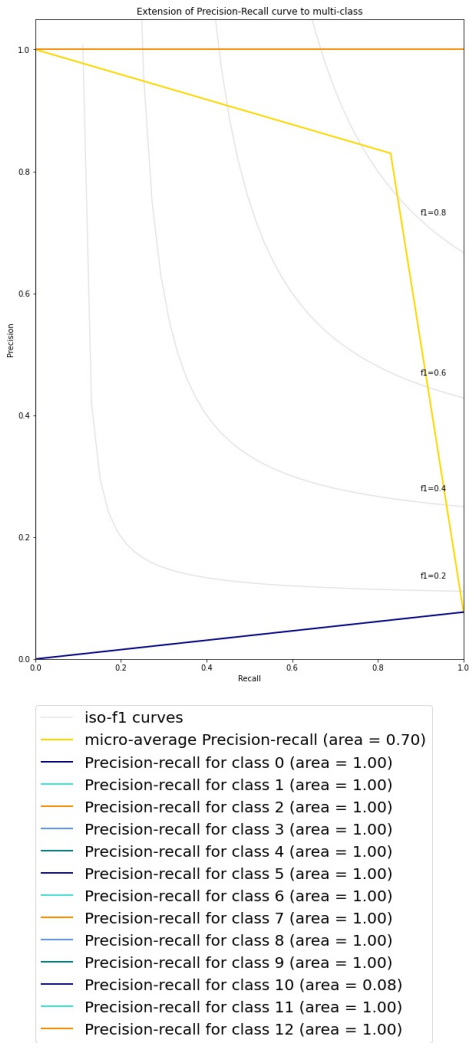


Figure 24: PR value obtained for each class from result of KNN trained on imbalanced dataset

score of all classess and Figure 27 shows PR value of each class for results obtained by testing KNN model, trained on dataset which is balanced using SMOTE analysis.

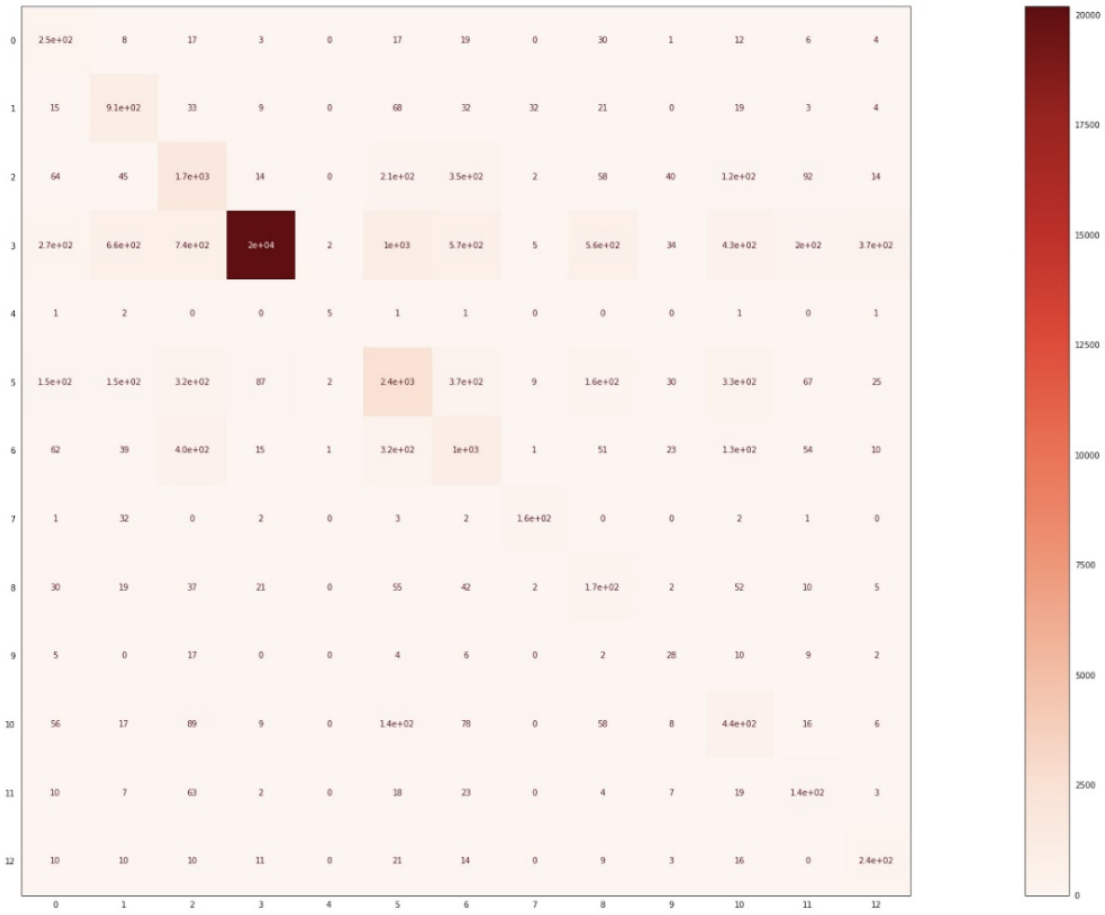


Figure 25: Confusion matrix obtained from results of KNN trained on dataset balanced using SMOTE analysis

We perform undersampling on dataset for balancing and achieve accuracy of 60.32%. Figure 28 shows confusion matrix, Figure 29 shows micro averaged precision score of all classes and Figure 30 shows PR value of each class for results obtained by testing KNN model, trained on dataset which is balanced using undersampling.

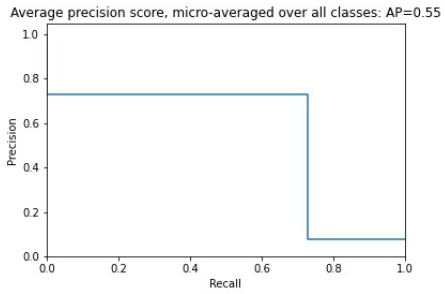


Figure 26: Micro averaged precision score obtained from results of KNN trained on dataset balanced using SMOTE analysis

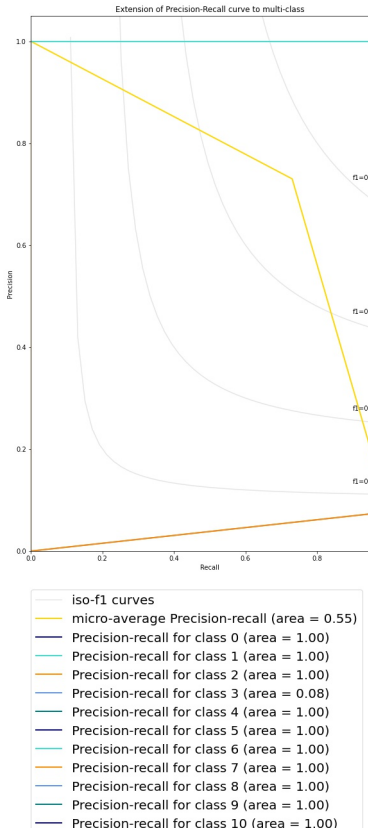


Figure 27: PR value obtained for each class from result of KNN trained on dataset balanced using SMOTE analysis

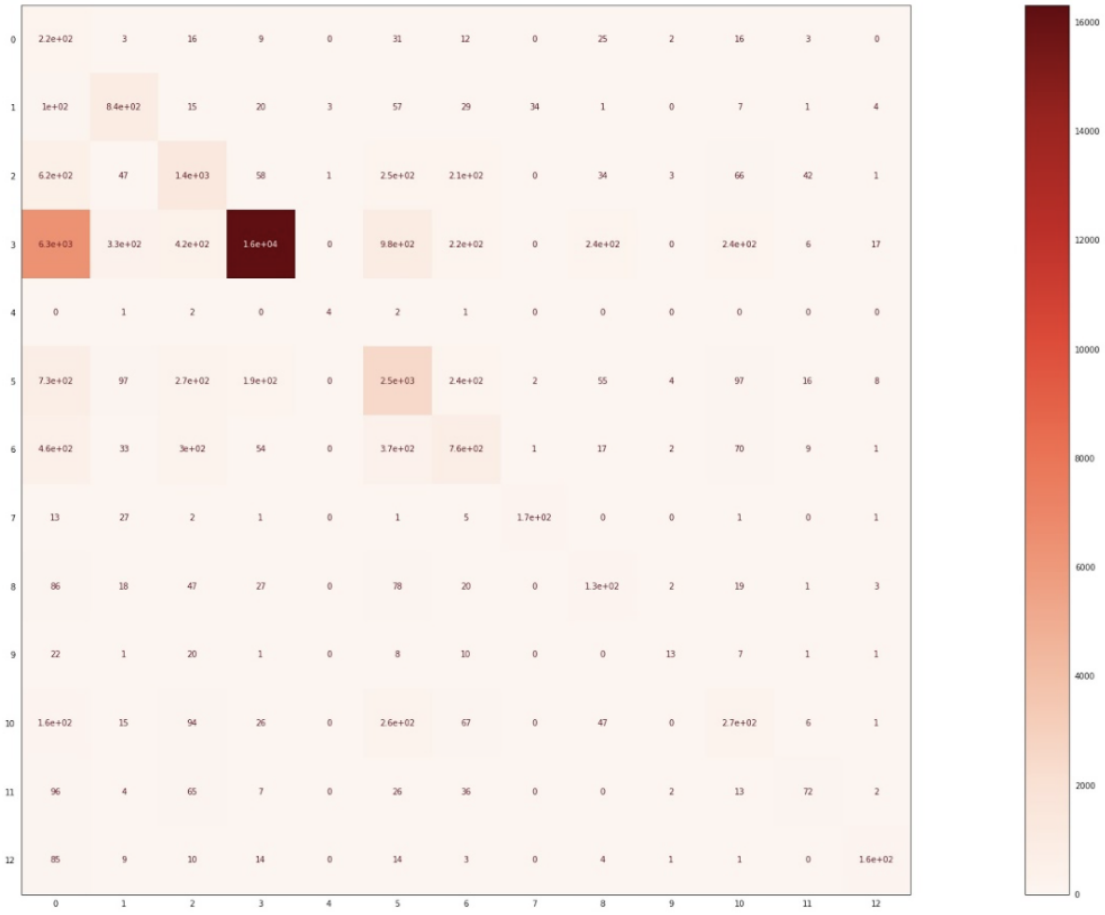


Figure 28: Confusion matrix obtained from results of KNN trained on dataset balanced using under sampling

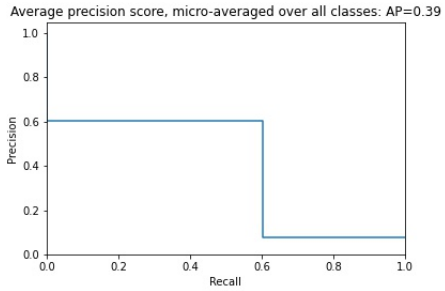


Figure 29: Micro averaged precision score obtained from results of KNN trained on dataset balanced using under sampling

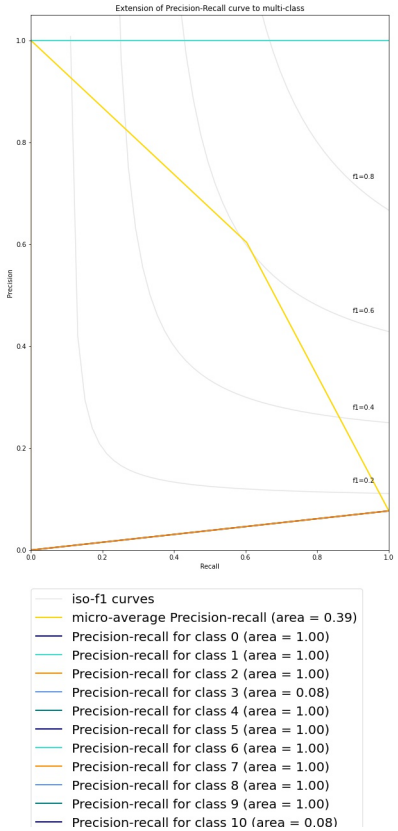


Figure 30: PR value obtained for each class from result of KNN trained on dataset balanced using under sampling

6.2.1.3 Multinomial Naïve Bayes (MNB)

Naïve Bayes is another machine learning algorithm which is highly used by researchers for text classification problems. Naive Bayes gives accuracy of 82.46% on imbalanced dataset.

Figure 31 shows confusion matrix, Figure 32 shows micro averaged precision score of all classes, Figure 33 shows PR value for each class from results obtained by testing model on imbalanced dataset.

As class is highly imbalanced, we balance the dataset using various data balancing technique.

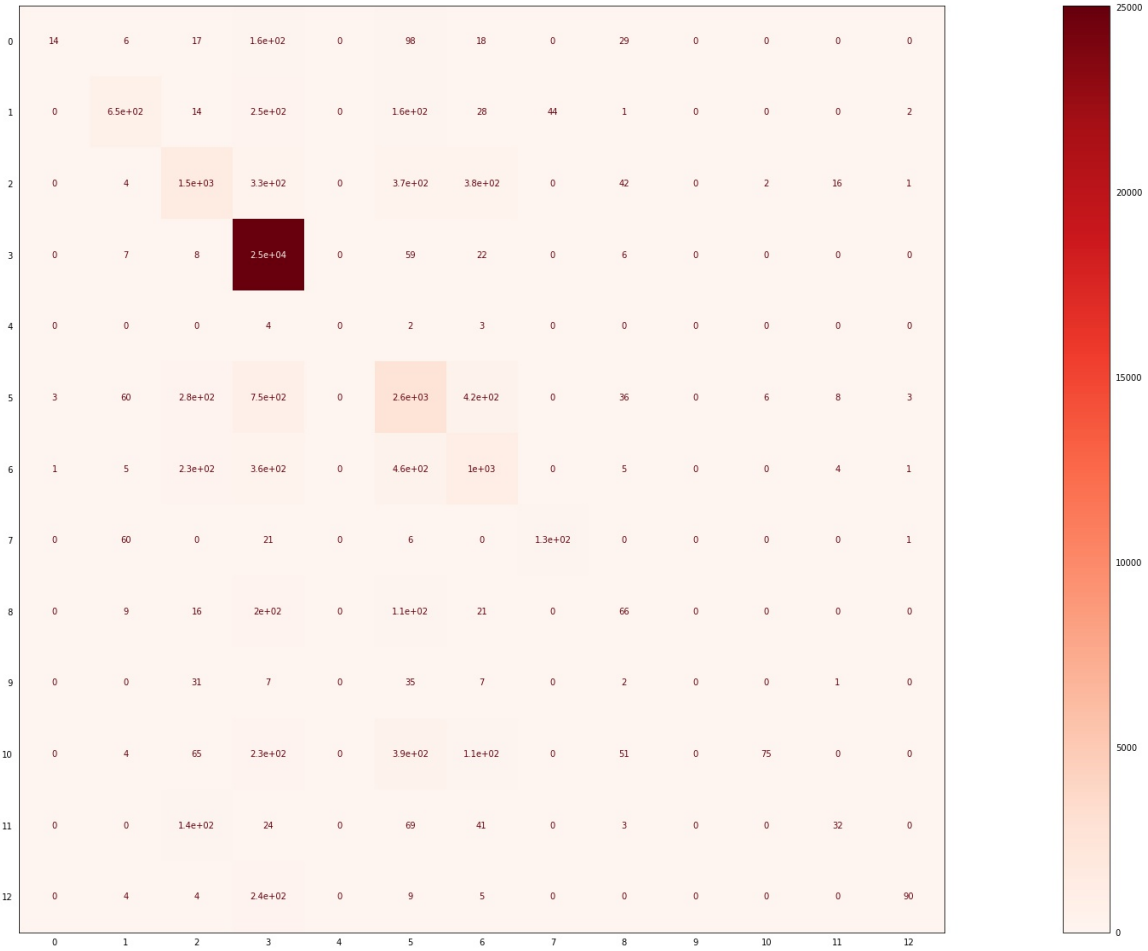


Figure 31: Confusion matrix obtained from results of Multinomial Naive Bayes for imbalanced dataset

We perform SMOTE analysis on dataset for upsampling and achieve accuracy of 81.36%. Figure 34 shows confusion matrix, Figure 35 shows micro averaged precision score of all classes and Figure 36 shows PR value of each class for results obtained by testing MNB model, trained on dataset which is balanced using SMOTE analysis.

We perform undersampling on dataset for balancing and achieve accuracy of 83.49%. Figure 37 shows confusion matrix, Figure 38 shows micro averaged precision score of all classes and Figure 39 shows PR value of each class for results obtained by testing MNB model, trained on dataset which is balanced using undersampling.

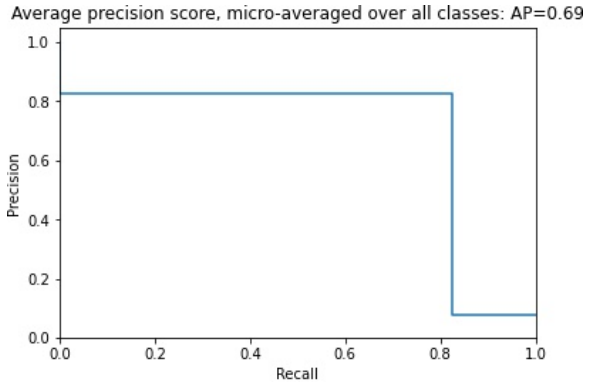


Figure 32: Micro averaged precision score obtained from results of Multinomial Naive Bayes for imbalanced dataset

Table 9: Logistic Regression Experiments on Doc2Vec vectors

Solver	Accuracy(%)
sag	66.36
saga	66.36
newton-cg	76.76
lbfgs	66.37

6.2.2 Doc2Vec

In Doc2Vec approach, we treat production description of each advertisement as a separate document. Using pre trained Doc2Vec model, we form a vector of 300 dimensions for each document. Using that vectors as feature, we train logistic regression model.

We experiment with various solvers for logistic regression. Table 9 shows accuracy of logistic regression for each solver.

As evident from Table 9, logistic regression with solver sag, saga and lbfgs gave accuracy around 66% on test data. Further analysis using confusion matrix, we find out that model trained using solver sag, saga and lbfgs predict only one class which is Drugs and Chemicals for all inputs. Since dataset is highly biased towards class

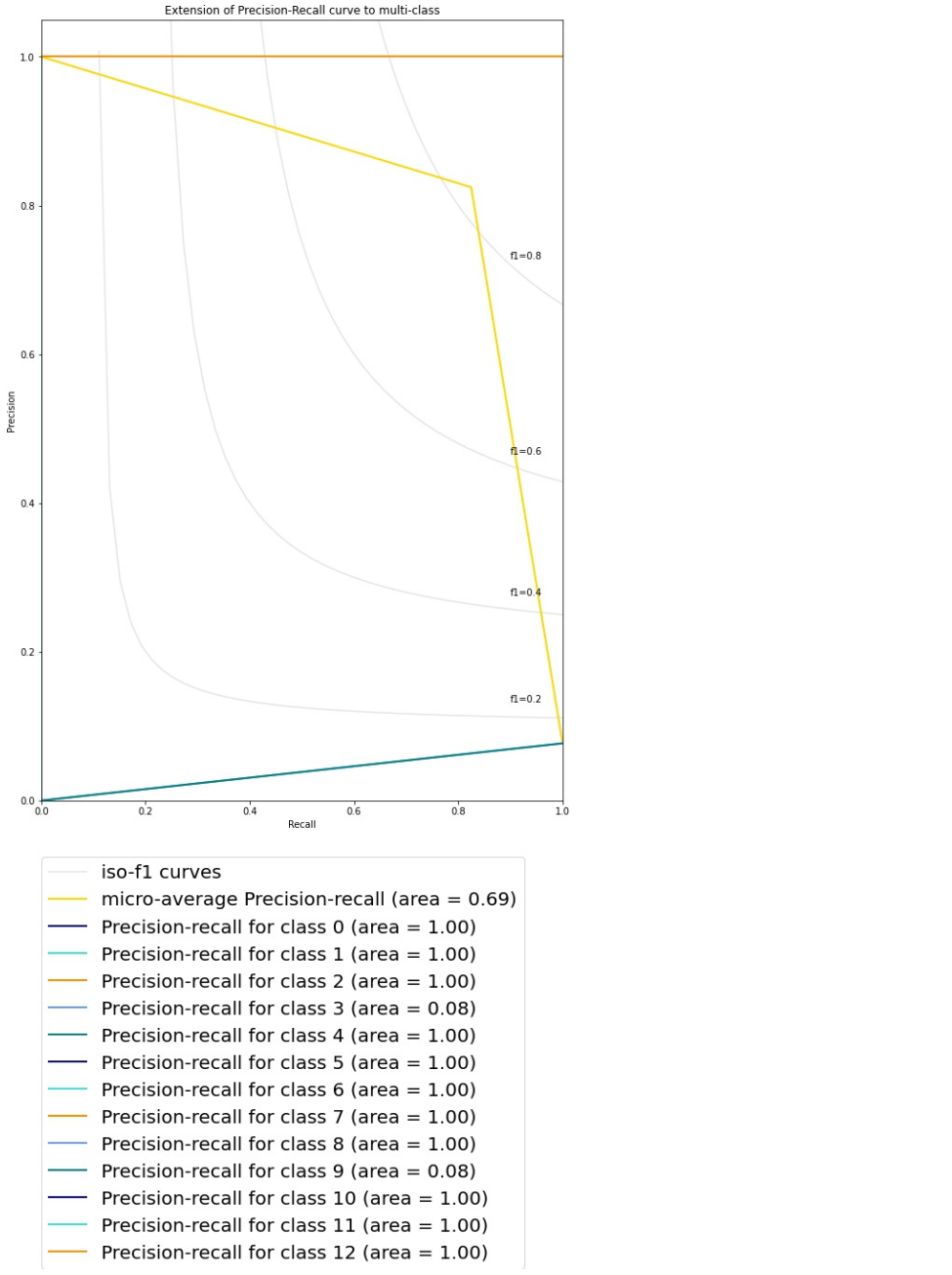


Figure 33: PR value obtained for each class from result of Multinomial Naive Bayes for imbalanced dataset

Drugs and Chemicals, model can not be trained and predict only one class.

Logistic regression with solver newton-cg give accuracy of 76.76% on test data. Further analysis is done on test results obtained from model trained using solver

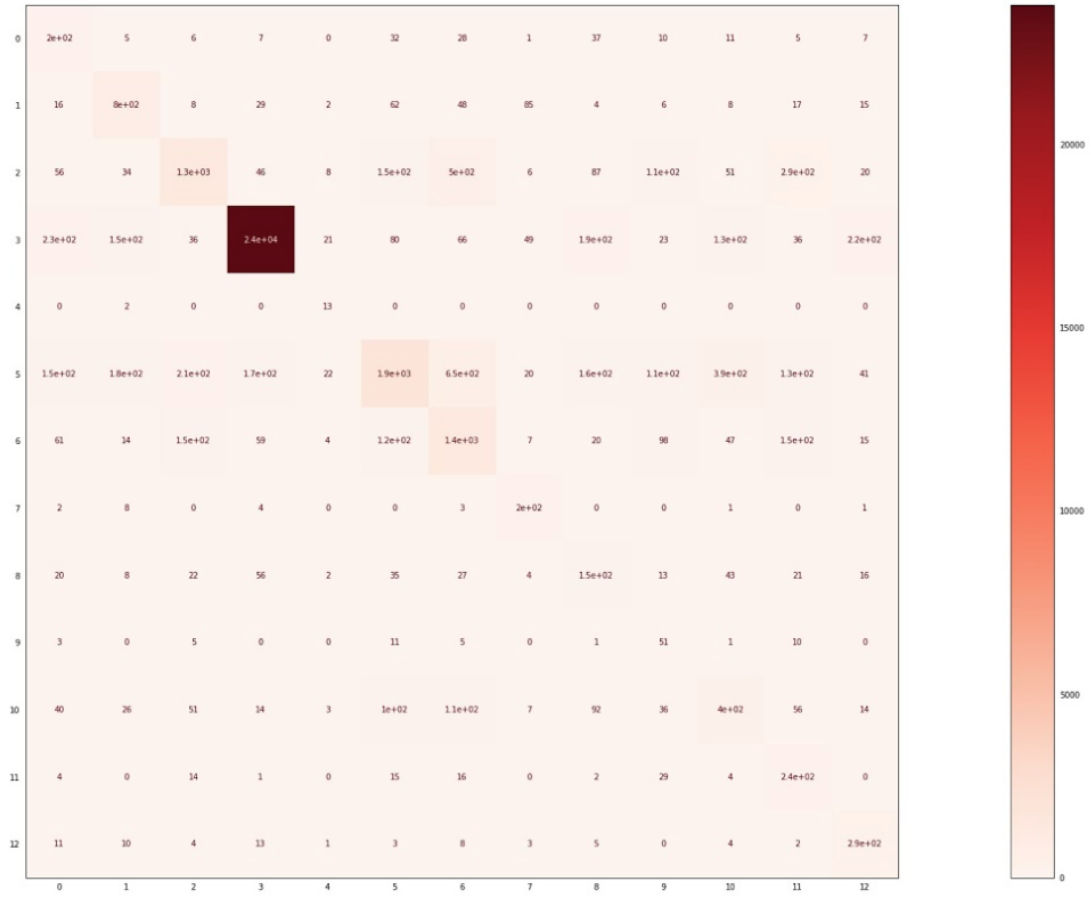


Figure 34: Confusion matrix obtained from results of MNB for dataset balanced using SMOTE analysis

Figure 35: Micro averaged precision score obtained from results of MNB for dataset balanced using SMOTE analysis

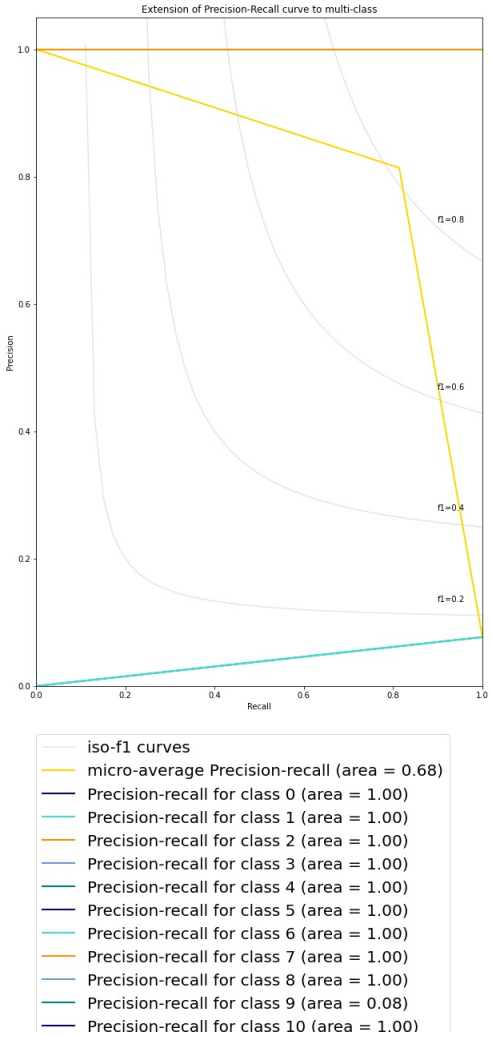


Figure 36: PR value obtained for each class from result of MNB for dataset balanced using SMOTE analysis

newton-cg.

Figure 40 shows confusion matrix, Figure 41 shows micro averaged precision score of all classes, Figure 42 shows PR value for each class from results obtained by testing logistic regression model trained on imbalanced dataset.

As class is highly imbalanced, we balance the dataset using various data balancing

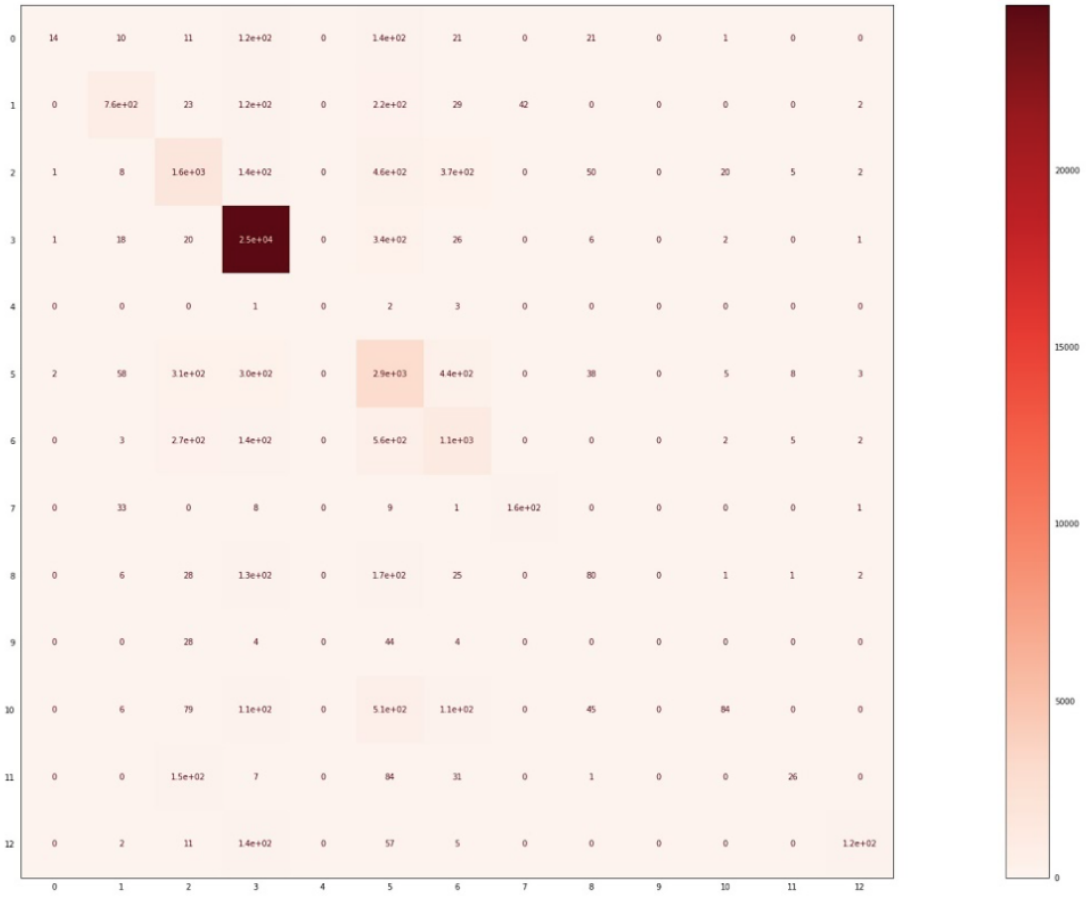


Figure 37: Confusion matrix obtained from results of MNB for dataset balanced using under sampling

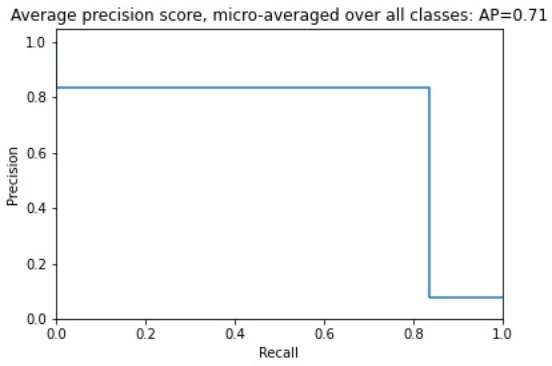


Figure 38: Micro averaged precision score obtained from results of MNB for dataset balanced using under sampling

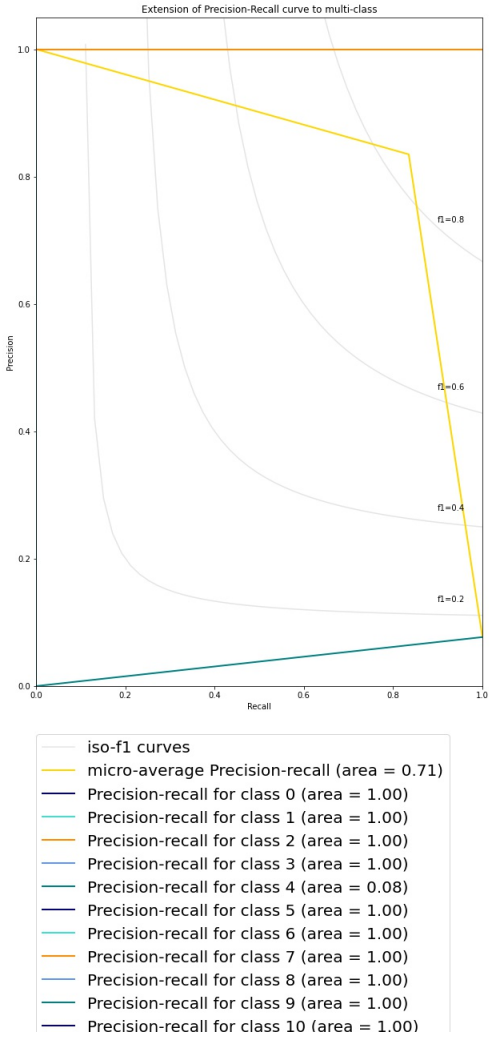


Figure 39: PR value obtained for each class from result of MNB for dataset balanced using under sampling

technique.

We perform SMOTE analysis on dataset for upsampling and achieved accuracy of 57.67%. Figure 43 shows confusion matrix Figure 44 shows micro averaged precision score of all classes and Figure 45 shows PR value of each class for results obtained by testing logistic regression model, trained on dataset which is balanced using SMOTE

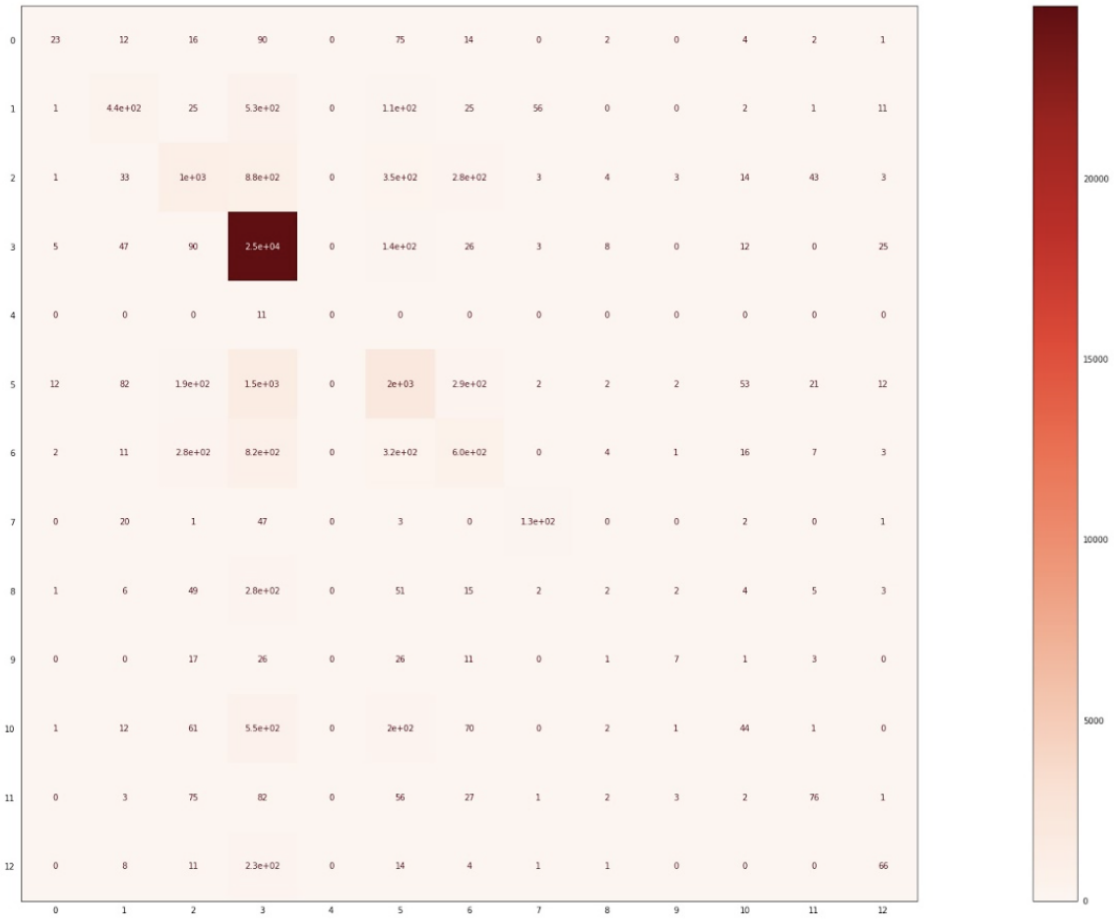


Figure 40: Confusion matrix obtained from results of logistic regression trained on imbalanced dataset formed using Doc2Vec

analysis.

We perform undersampling on dataset for balancing and achieve accuracy of 73.50%. Figure 46 shows confusion matrix, Figure 47 shows micro averaged precision score of all classes and Figure 48 shows PR value of each class for results obtained by testing logistic regression model, trained on dataset which is balanced using undersampling.

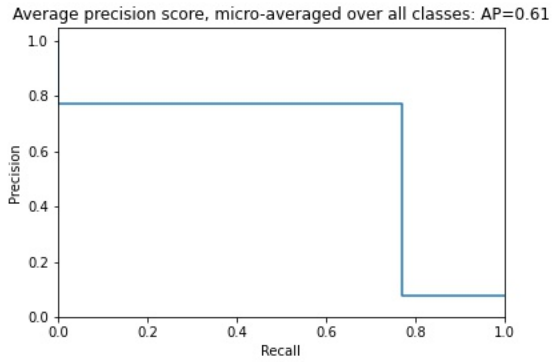


Figure 41: Micro averaged precision score obtained from results of logistic regression trained on imbalanced dataset formed using Doc2Vec

6.2.3 BERT

In this approach, we use pre trained bert base uncased model. We provide tokenized product description as input to BERT model to generate 769 dimension vectors for each product description. These vectors are used to train logistic regression model.

Logistic regression with solver newton-cg give accuracy of 82.98% on test data after getting trained on vectors generated from BERT.

Figure 49 shows confusion matrix, Figure 50 shows micro averaged precision score of all classes, Figure 51 shows PR value for each class from results obtained by testing logistic regression model on imbalanced dataset formed using BERT.

6.3 Comparison of results

In this section we compare results obtained by training and testing machine learning models using dataset. We compare various model using two measures. They are accuracy and average precision score.

6.3.1 Comparison of results using Accuracy

Table 10 shows accuracy of different model.

As evident from Table 10 SVM outperforms all other models when compared

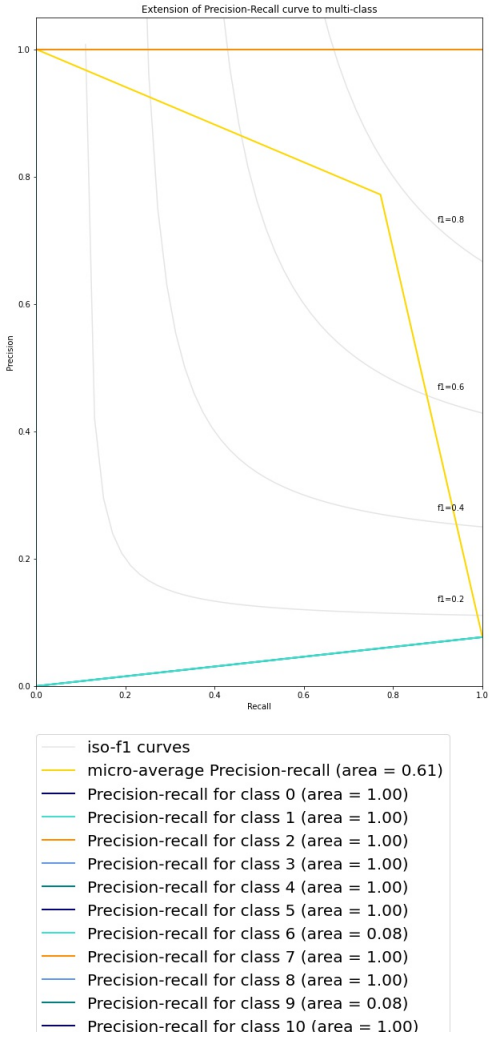


Figure 42: PR value obtained for each class from results of logistic regression trained on imbalanced dataset formed using Doc2Vec

on the basis of accuracy. Balancing data degrades performance primarily due to overfitting and loosing information when done with SMOTE and under sampling respectively. Only Naive Bayes gives better accuracy when data is balanced using under sampling.

Doc2Vec embeddings and logistic regression gives poor result when compared

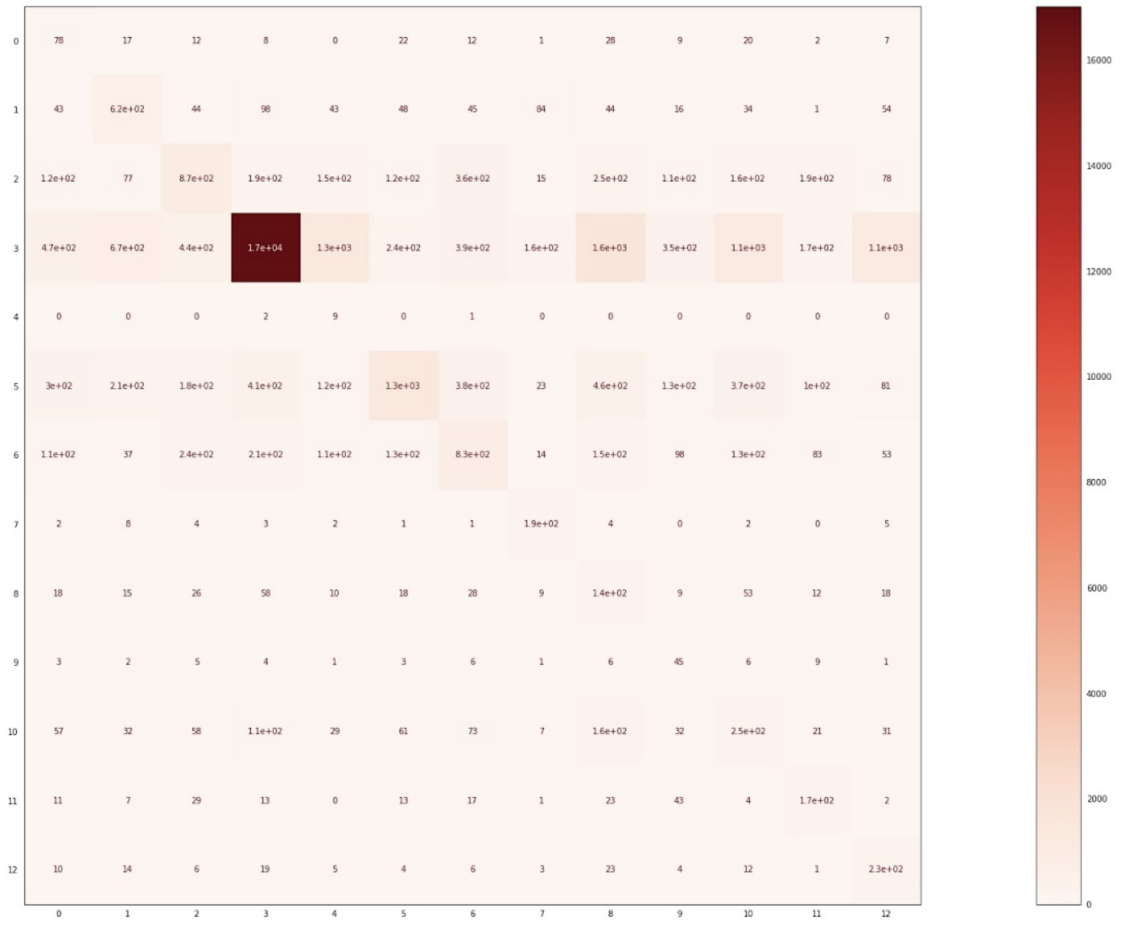


Figure 43: Confusion matrix obtained from results of logistic regression trained on dataset balanced using SMOTE analysis and formed using Doc2Vec

with TF-IDF and SVM. However, BERT and logistic regression gives result close to TF-IDF AND SVM.

6.3.2 Comparison of results using Average Precision Score

Here we have computed micro averaged precision score of various model and done comparison. Table 11 shows accuracy of different model.

As evident from Table 11 SVM outperforms all other models when compared on the basis of average precision score. Balancing data degrades performance for all most

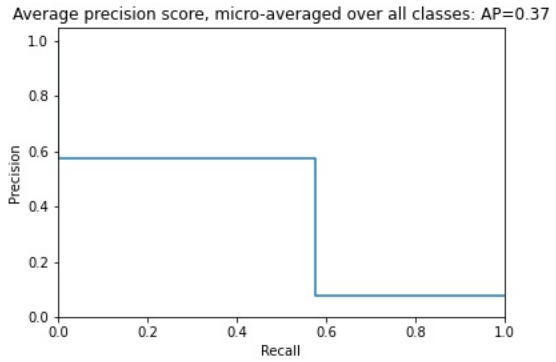


Figure 44: Micro averaged precision score obtained from results of logistic regression trained on dataset balanced using SMOTE analysis and formed using Doc2Vec

Table 10: Comparison of various machine learning models based on accuracy

Vectors Generation	Machine Learning Model	Dataset	Accuracy(%)
TF-IDF	SVM	Imbalanced	86.50
TF-IDF	KNN	Imbalanced	78.49
TF-IDF	KNN	Balanced using SMOTE	73.01
TF-IDF	KNN	Balanced using under sampling	60.32
TF-IDF	Naive Bayes	Imbalanced	82.46
TF-IDF	Naive Bayes	Balanced using SMOTE	81.36
TF-IDF	Naive Bayes	Balanced using under sampling	83.49
Doc2Vec	Logistic Regression	Imbalanced	76.76
Doc2Vec	Logistic Regression	Balanced using SMOTE	57.67
Doc2Vec	Logistic Regression	Balanced using under sampling	73.50
BERT	Logistic Regression	Imbalanced	82.98

all models.

Doc2Vec does not give better result but BERT gives result similar to TF-IDF and SVM. Table 12 and Table 13 show comparison between TF-IDF and BERT for each product.

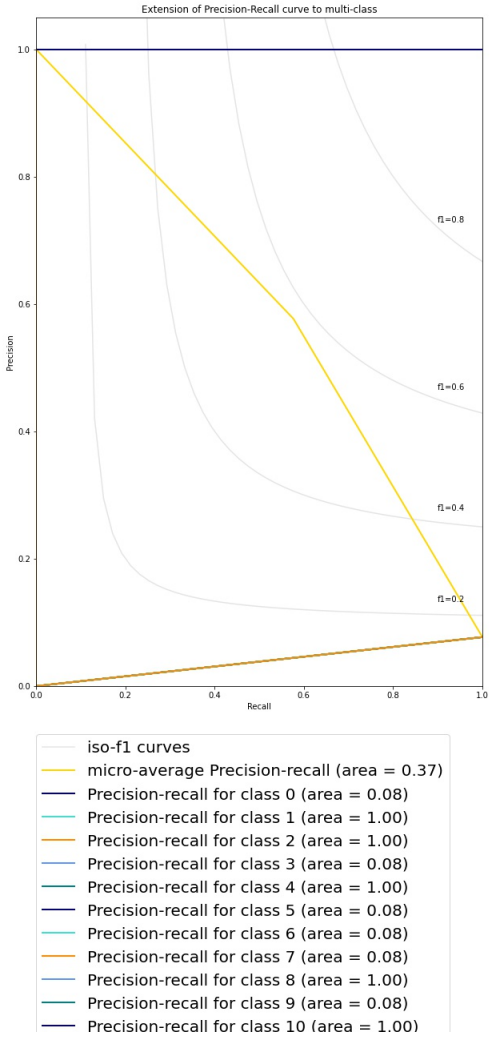


Figure 45: PR value obtained for each classfrom results of logistic regression trained on dataset balanced using SMOTE analysis and formed using Doc2Vec

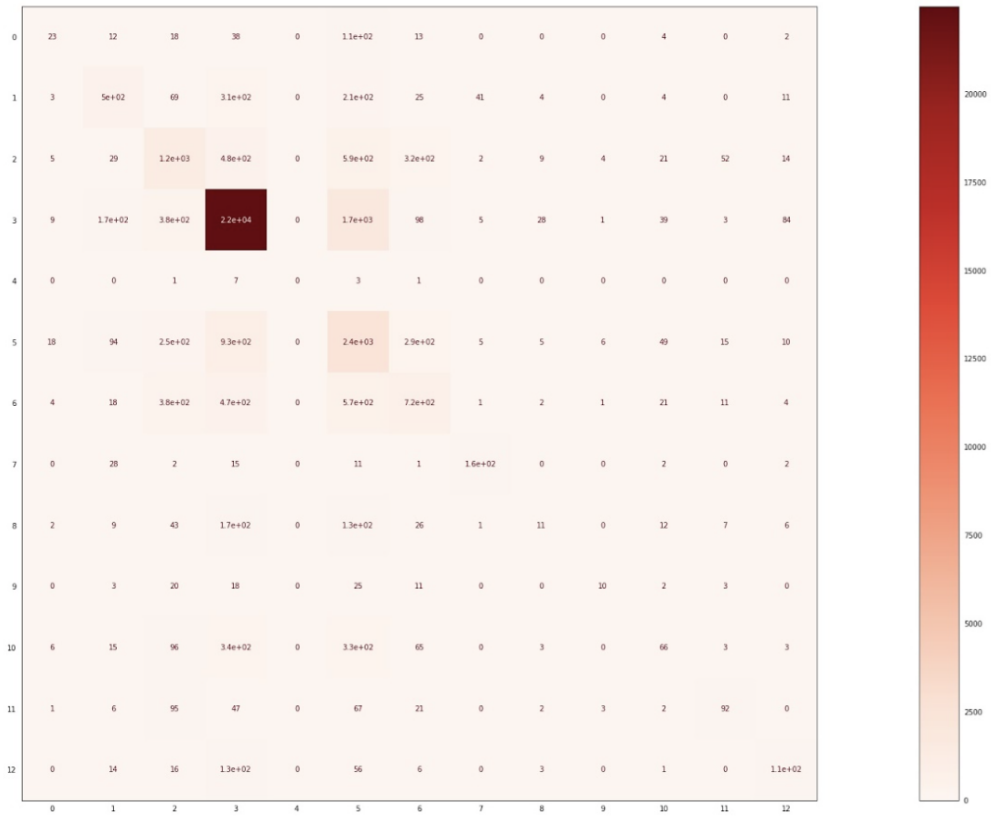


Figure 46: Confusion matrix obtained from results of logistic regression trained on dataset balanced using undersampling and formed using Doc2Vec

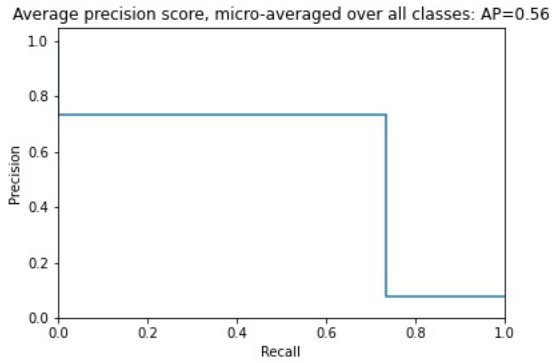


Figure 47: Micro averaged precision score obtained from results of logistic regression for dataset balanced using under sampling and formed using Doc2Vec

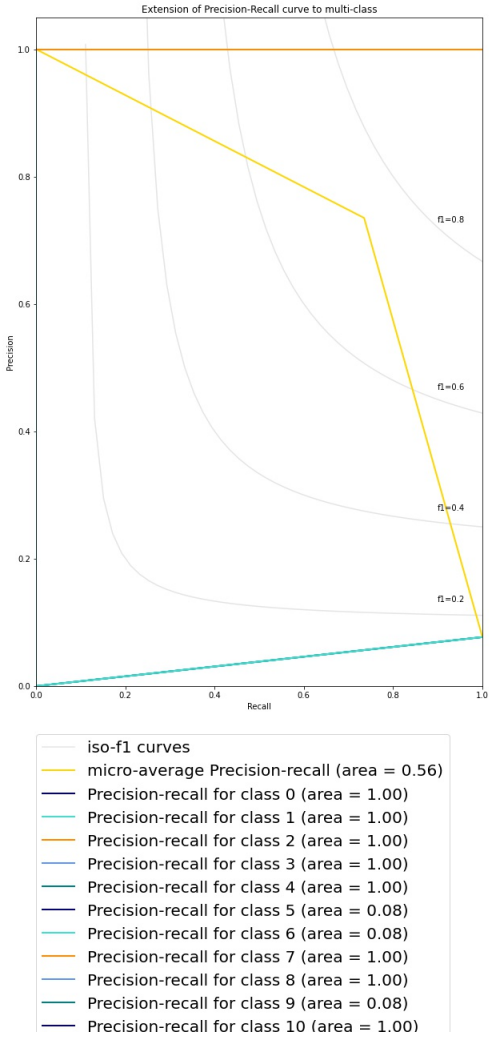


Figure 48: PR value obtained for each class from result of logistic regression for dataset balanced using under sampling and formed using Doc2Vec

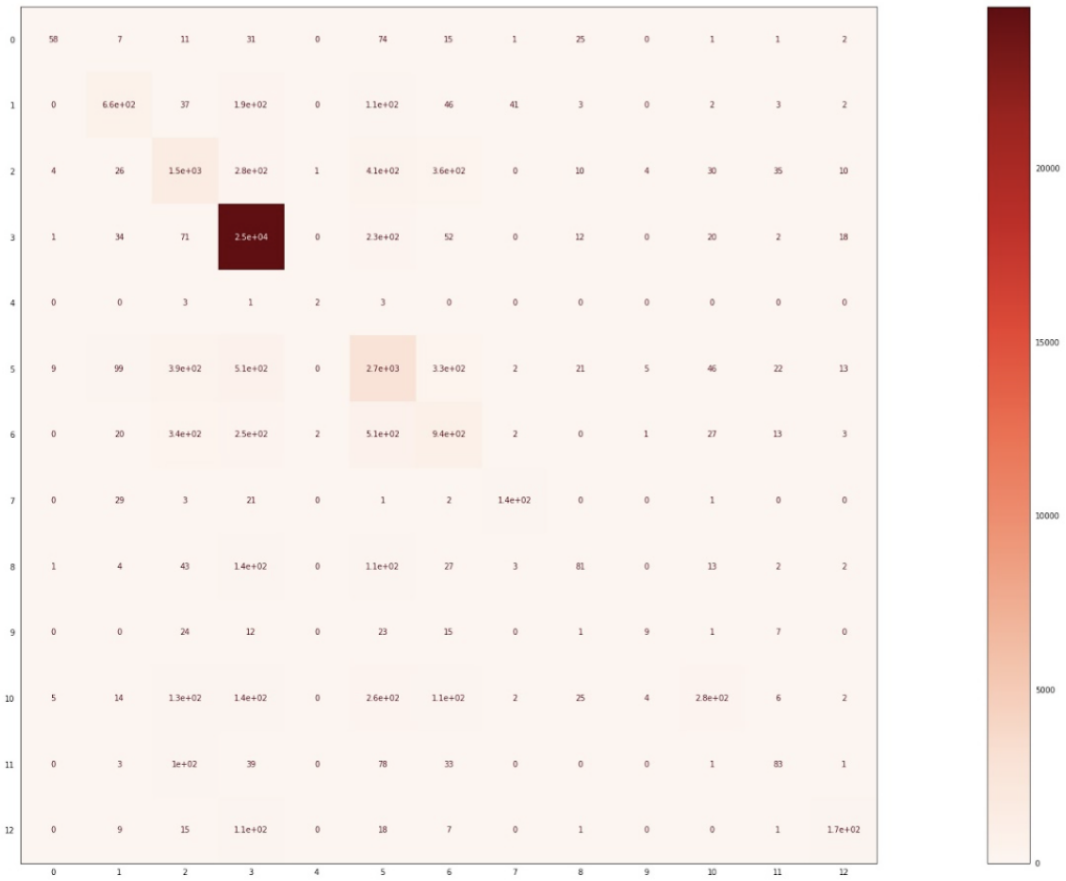


Figure 49: Confusion matrix obtained from results of logistic regression trained using imbalanced dataset and formed using BERT

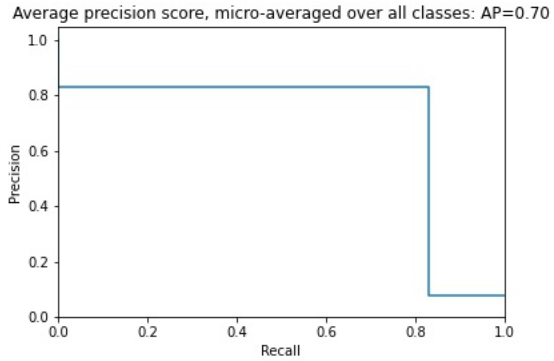


Figure 50: Micro averaged precision score obtained from results of logistic regression trained using imbalanced dataset and formed using BERT

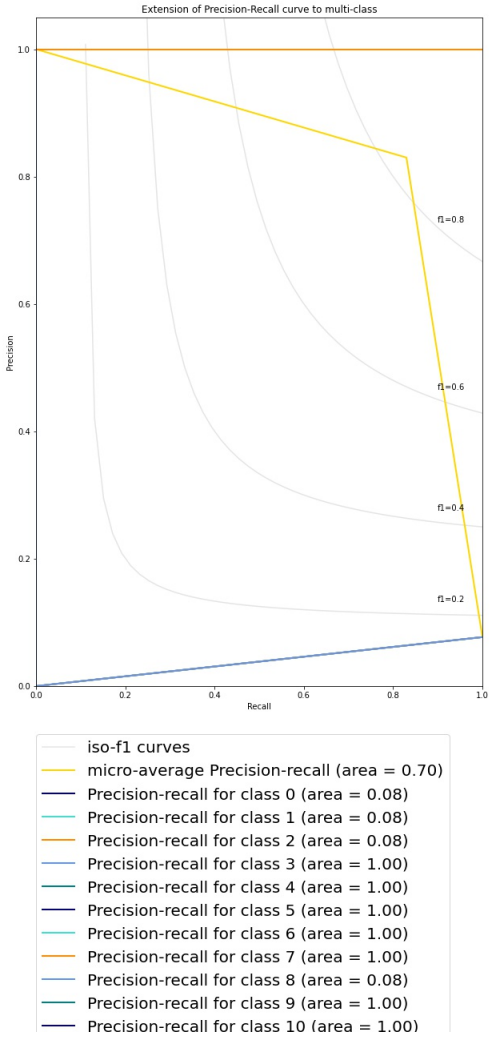


Figure 51: PR value obtained for each class from result of logistic regression trained using imbalanced dataset and formed using BERT

Table 11: Comparison of various machine learning models based on average precision score

Vector Generation	Machine Learning Model	Dataset	AP Score
TF-IDF	SVM	Imbalanced	0.75
TF-IDF	KNN	Imbalanced	0.70
TF-IDF	KNN	Balanced using SMOTE	0.55
TF-IDF	KNN	Balanced using under sampling	0.39
TF-IDF	Naive Bayes	Imbalanced	0.69
TF-IDF	Naive Bayes	Balanced using SMOTE	0.68
TF-IDF	Naive Bayes	Balanced using under sampling	0.71
Doc2Vec	Logistic Regression	Imbalanced	0.61
Doc2Vec	Logistic Regression	Balanced using SMOTE	0.37
Doc2Vec	Logistic Regression	Balanced using under sampling	0.56
BERT	Logistic Regression	Imbalanced	0.70

Table 12: PR Value for each product from result obtained by testing SVM model

Category	Precision Recall value
Carded Items	1
Counterfeit Items	1
Digital Products	0.08
Drugs & Chemicals	1
Electronics	1
Fraud	1
Guides & Tutorial	1
Jewels & Gold	1
Other Listings	1
Security & Hosting	1
Services	1
Software & Malware	1
Weapons	1

Table 13: PR Value for each product from result obtained by testing logistic regression model trained using BERT vectors

Category	Precision Recall value
Carded Items	0.08
Counterfeit Items	0.08
Digital Products	0.08
Drugs & Chemicals	1
Electronics	1
Fraud	1
Guides & Tutorial	1
Jewels & Gold	1
Other Listings	1
Security & Hosting	0.08
Services	1
Software & Malware	1
Weapons	1

CHAPTER 7

Conclusion and Future work

In this project, we experiment with the state of art document embeddings, like Doc2Vec and BERT, in discovering knowledge for the darknet domain. We combine data from Alphabay road and Hansa marketplace networks and enhance it manually by adding labels to form a new dataset. As a first step, we perform data pre-processing to clean the data and to make it more accurate. In one of our approaches, we form vectors of the data using TF-IDF, and train using SVM, KNN, and Naïve Bayes algorithm. In another approach, we use the models of Doc2Vec and BERT to create embeddings and train with logistic regression algorithms. By analyzing our result, we find that Doc2Vec embeddings are lacking in the final results, but BERT embeddings give promising results when compared against TF-IDF vectorization. When the logistic regression model is trained using the BERT vectors, the results are better than the KNN models that are trained using TF-IDF vectors. The logistic regression model with BERT vectors gives results very close to SVM and Naïve Bayes model with TF-IDF vectors. Overall, the SVM model trained using TF-IDF vectors gives the best result in terms of accuracy and micro averaged precision score. We also experiment with various data balancing techniques on the imbalanced dataset and train the model using balanced data. However, balanced data give poor results on the training model.

A few categories are very similar, and it makes it very difficult even for humans to distinguish from one category to another. For example, category "Fraud" and "carded items". For future work, we can define a metric to work around this issue. Apart from that, another issue we encountered is that vectors generated using Doc2Vec and BERT are of very high dimensions. In future work, we can apply some dimensionality reduction algorithms like PCA on the vectors and train the model using that vectors.

LIST OF REFERENCES

[1] ‘‘Confusion matrix example.’’ [Online]. Available: https://www.jeremyjordan.me/evaluating-a-machine-learning-model

[2] ‘‘Pr curve example.’’ [Online]. Available: https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

[3] ‘‘word2vecdoc2vec architecture.’’ [Online]. Available: https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

[4] ‘‘Bert architecture.’’ [Online]. Available: http://jalammar.github.io/illustrated-bert/

[5] ‘‘stopwords example.’’ [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html

[6] ‘‘wordnormalization example.’’ [Online]. Available: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

[7] M. Stamp, *Introduction to machine learning with applications in information security*. CRC Press, 2017.

[8] ‘‘Knn.’’ [Online]. Available: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

[9] P. Domingos and M. Pazzani, ‘‘On the optimality of the simple bayesian classifier under zero-one loss,’’ *Machine learning*, vol. 29, no. 2, pp. 103--130, 1997.

[10] ‘‘logisticregression.’’ [Online]. Available: https://towardsdatascience.com/logistic-regression-explained-9ee73cede081

[11] ‘‘History of dark web.’’ [Online]. Available: https://www.soscanhelp.com/blog/history-of-the-dark-web

[12] N. Christin, ‘‘Traveling the silk road: A measurement analysis of a large anonymous online marketplace,’’ in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 213--224.

[13] G. Branwen, ‘‘Silk road 1: Theory & practice,’’ 2011.

[14] U. Noor, Z. Rashid, and A. Rauf, ‘‘A survey of automatic deep web classification techniques,’’ *International Journal of Computer Applications*, vol. 19, no. 6, pp. 43--50, 2011.

[15] A. Biryukov, I. Pustogarov, F. Thill, and R.-P. Weinmann, ''Content and popularity analysis of tor hidden services,'' in *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2014, pp. 188--193.

[16] M. Graczyk and K. Kinningham, ''Automatic product categorization for anonymous marketplaces,'' Technical report, Stanford University, Tech. Rep., 2015.

[17] D. Moore and T. Rid, ''Cryptopolitik and the darknet,'' *Survival*, vol. 58, no. 1, pp. 7--38, 2016.

[18] S. Ghosh, A. Das, P. Porras, V. Yegneswaran, and A. Gehani, ''Automated categorization of onion sites for analyzing the darkweb ecosystem,'' in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1793--1802.

[19] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. de Paz, ''Classifying illegal activities on tor network based on web textual contents,'' in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 35--43.

[20] M. W. Al-Nabki, E. Fidalgo, E. Alegre, and L. Fernández-Robles, ''Torank: Identifying the most influential suspicious domains in the tor network,'' *Expert Systems with Applications*, vol. 123, pp. 212--226, 2019.

[21] P.-Y. Du, M. Ebrahimi, N. Zhang, H. Chen, R. A. Brown, and S. Samtani, ''Identifying high-impact opioid products and key sellers in dark net marketplaces: An interpretable text analytics approach,'' in *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2019, pp. 110--115.

[22] J. Li, Q. Xu, N. Shah, and T. K. Mackey, ''A machine learning approach for the detection and characterization of illicit drug dealers on instagram: model evaluation study,'' *Journal of medical Internet research*, vol. 21, no. 6, p. e13803, 2019.

[23] S. Shan and R. Sankaranarayana, ''Behavioral profiling of darknet marketplace vendors,'' *https://www.gwern.net/*, 2020.

[24] S. Jeziorowski, M. Ismail, and A. Siraj, ''Towards image-based dark vendor profiling: An analysis of image metadata and image hashing in dark web marketplaces,'' in *IWSPA@CODASPY '20: Proceedings of the Sixth International Workshop on Security and Privacy Analytics, New Orleans, LA, USA, March 18, 2020*, R. M. Verma, L. Khan, and C. K. Mohan, Eds. ACM, 2020, pp. 15--22.

[25] L. Armona and D. Stackman, ''Learning darknet markets,'' *Federal Reserve Bank of New York mimeo*, 2014.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ‘‘Efficient estimation of word representations in vector space,’’ *arXiv preprint arXiv:1301.3781*, 2013.